WAVELET ANALYSIS IN COMPUTATIONAL FLUID DYNAMICS

by

Ali Berk Kahraman

B.S., Mechanical Engineering, Boğaziçi University, 2016

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Mechanical Engineering

Boğaziçi University

2018

WAVELET ANALYSIS IN COMPUTATIONAL FLUID DYNAMICS

APPROVED BY:

Assoc. Prof. Ali Ecder ..... (Thesis Supervisor)

Assist. Prof. Gülin Vardar .....

Assist. Prof. Hatice Mercan .....

DATE OF APPROVAL: 11.07.2018

ACKNOWLEDGEMENTS

In the beginning, I want to thank my mother, Mahmure Kahraman, for her endless efforts in raising me for years. The main reason I achieved to write this thesis is her dedication. She is heartfully appreciated.

I would like to thank my advisor Ali Ecder, who has guided me through my M.Sc. studies. His technical and scientific vision has paved the way for this thesis.

I would like to thank all my friends for their support. I appreciate the friendships of Ersin Yıldız and Tuğrul Yılmaz especially, their company was priceless.

This work was supported by The Scientific and Technological Research Council of Turkey(TÜBİTAK) 2210-A General Domestic Graduate Scholarship Programme.

ABSTRACT

WAVELET ANALYSIS IN COMPUTATIONAL FLUID DYNAMICS

Wavelets are compact functions in space and time which enable easy data compression through multi-resolution analysis. The compression is done by scattering different resolution levels of wavelets onto the domain, and then discarding the wavelets with small energies. Using these properties, wavelets can be an efficient and easily applied tool to construct an adaptive grid for the solution of a Partial Differential Equation (PDE) with local structures. In this work, wavelets are used in this manner of compressing the interpolated data, being combined with finite difference discretization to solve the PDE. To calculate the spatial derivatives of the compressed interpolation, algebraic polynomial fits and cubic splines are used on the irregular adaptive grid. These two approaches are compared with each other and finite differences on regular grids. Various problems in 1-D and 2-D are solved. As model problems, Poisson's Equation and Helmholtz Equation are solved with artificially created Gaussian Pulse as the solution. The results seemed to be in agreement in terms of the order of error with the known exact solutions of the model problems. A new application for wavelet optimized finite differences is also suggested. To that purpose, split-step (projection-correction) time scheme is implemented for the Navier-Stokes Equations governing infamous lid-driven cavity problem. The qualitative results seemed to be in agreement with other results in literature, however the method is observed to be not of any advantage for this problem as this problem does not have strong localized structures. PETSc framework provided the high-level tools, such as matrix and vector operations and linear solvers, and the work is conducted through this perspective.

ÖZET

HESAPLAMALI AKIŞKANLAR DİNAMİĞİNDE DALGACIK ANALİZİ

Dalgacıklar, çoklu çözünürlüklü analiz üzerinden kolaylıkla bilgi sıkıştırmasını sağlayan uzay ve zaman içerisinde kompakt fonksiyonlardır. Bu sıkıştırma, dalgacıkları değişik çözünürlük seviyelerinde alana yayıp, enerjisi düşük olan dalgacıkları çıkararak yapılır. Bu özellikler kullanılarak, dalgacıklar yerel yapılar barındıran kısmi diferansiyel denklemlerin çözümü için adaptif ağ inşasında etkili ve kolayca uygulanabilen bir gereç olabilir. Bu çalışmada dalgacıklar bu yönde kullanılarak enterpolasyonlu verinin sıkıştırılması ve sonlu farklar ayrıklaştırması birleştirilerek kısmi diferansiyel denklemler çözülmüştür. Sıkıştırılmış enterpolasyonda uzaysal türevlerin hesaplanması için düzensiz ağ üzerinde cebirsel polinom uydurması ve kübik yaklaşımlar kullanılmıştır. Bu iki yaklaşım birbiriyle ve düzenli ağda sonlu farklar yöntemiyle karşılaştırılmıştır. Bir boyutta ve iki boyutta değişik problemler çözülmüştür. Model problemler olarak, yapay olarak oluşturulmuş bir Gauss Titreşimi çözümü olan Poisson Denklemi ve Helmholtz Denklemi çözülmüştür. Sonuçların, bilinen kesin çözüme karşı iyi bir hata mertebesinde olduğu görülmüştür. Dalgacıkla optimize edilmiş sonlu farklar ayrıklaştırması için yeni bir uygulama da önerilmiştir. Bu amaç için, bölünmüş adım (yansıtma ve düzeltme) zaman şeması Navier-Stokes denklemi tarafından yönetilen ünlü kapak tahrikli boşluk problemine uygulanmıştır. Nitel sonuçların literatürdeki diğer sonuçlarla uyumlu olduğu gözlemlenmiş, ancak metodun bu problem özelinde problemin güçlü yerel yapılara sahip olmaması nedeniyle bir avantajı olmadığı gözlemlenmiştir. Çatı olarak yüksek seviyedeki gereçleri, matris ve vektör operasyonları ve lineer sistem çözücüleri gibi, sağlaması için PETSc kullanılmıştır ve bu çalışma bu perspektiften yürütülmüştür.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF SYMBOLS . . . xiii
LIST OF ACRONYMS/ABBREVIATIONS . . . xvi
1. INTRODUCTION . . . 1
2. WAVELET ANALYSIS . . . 6
2.1. 1-D Multi-Resolution Wavelet Analysis . . . 7
2.1.1. 1-D Lifting Scheme . . . 9
2.1.2. 1-D Compression of Interpolation Function . . . 10
2.1.3. Implementation of 1-D Wavelet Transform . . . 11
2.1.3.1. Regular Grid . . . 12
2.1.3.2. Irregular Grid . . . 13
2.2. Spatial Differentiation on Irregular Grid in 1-D . . . 15
2.2.1. Polynomial Fit Approach . . . 15
2.2.2. Cubic Spline Approach . . . 16
2.3. 2-D Multi-Resolution Wavelet Analysis . . . 18
2.3.1. 2-D Lifting Scheme . . . 20
2.3.2. 2-D Compression of Interpolation Function . . . 21
2.3.2.1. Minimum Set Construction . . . 23
2.4. Spatial Differentiation on Irregular Grid in 2-D . . . 25
2.4.1. An Important Remark . . . 28
3. NUMERICAL PROCEDURE AND SOLVED PROBLEMS . . . 32
3.1. 1-D Poisson Equation . . . 33
3.1.1. Moving Local Structure . . . 33
3.1.2. Disappearing and Reappearing Local Structure . . . 34
3.2. 2-D Poisson Equation . . . 34
3.3. 2-D Helmholtz Equation . . . 35

3.4. 2-D Isothermal Navier-Stokes Equation 36
3.4.1. Time Integration Scheme 36
3.4.1.1. Poisson Neumann Equation 38
3.4.2. Lid Driven Cavity 40
4. RESULTS AND DISCUSSIONS 42
4.1. 1-D Poisson Equation 42
4.1.1. Moving Local Structure 42
4.1.2. Disappearing and Reappearing Local Structure 42
4.2. 2-D Poisson Equation 46
4.3. 2-D Helmholtz Equation 52
4.4. 2-D Isothermal Navier-Stokes Equations 56
4.4.1. Lid Driven Cavity 56
5. CONCLUSIONS 62
REFERENCES 65

LIST OF FIGURES

Figure 2.1. Different kinds of wavelets, (a) Daubechies' 2 Wavelet, (b) Autocorrelation of Daubechies' 2, (c) Mexican Hat Wavelet. 6

Figure 2.2. The nested grid decomposed to its level for 1-D 8

Figure 2.3. Forward wavelet coefficient calculation. 12

Figure 2.4. Forward basis function coefficient calculation. 12

Figure 2.5. Forward wavelet transformation. 13

Figure 2.6. Forward wavelet transformation on an irregular grid. 14

Figure 2.7. The nested grid decomposed to its levels for 2-D. ■ = Basis Function of level $j = 2$, ♦ = Wavelets of level $j = 2$, · = Wavelets of level $j = 3$. 19

Figure 2.8. First option for adjacent zone creation in 2-D. · = adjacent zone in x, ×=adjacent zone in y. 22

Figure 2.9. Second option for adjacent zone creation in 2-D. · = adjacent zone created by the wavelet on x, × = adjacent zone created by the wavelet on y. 23

Figure 2.10. Minimum set points for three types of wavelets, (a) is for type 1, (b) is for type 2 and (c) is for type 3. ■ = the originally active grid point,○ and • = the minimum set points. 24

Figure 2.11. Minimum set construction. 24

Figure 2.12. An example minimum set grid where the square is the originally active wavelet. 25

Figure 2.13. A centered stencil. Active points are denoted with ·, the differentiation point is marked with ∘ and the stencil points are marked with □ in addition to ·. 26

Figure 2.14. The first method for completing the stencils in the grid. 27

Figure 2.15. The second method for completing the stencils in the grid. 28

Figure 2.16. The comparison of the minimum set, first method and second method of completion of the grid. ■ = active wavelet, · = minimum set points, × = points added by first method, □ = points added by second method. 29

Figure 2.17. The sample function and the adapted grid. 30

Figure 2.18. The error for the Laplacian of the sample function on the adapted grid. 31

Figure 3.1. Time integration scheme. 32

Figure 4.1. The time evolution of the solution (a,c,e) and the grid (b,d,f) for right travelling Gaussian Pulse at $t = 0$, $t = 0.25$, $t = 0.5$ from top to bottom. 43

Figure 4.2. The error comparison between polynomial fit differentiation, cubic spline differentiation and regular finite differences for right travelling Gaussian Pulse. 44

Figure 4.3. The time evolution of the solution (a,c,e) and the grid (b,d,f) for disappearing and reappearing Gaussian Pulse at $t = 0$, $t = 1.0$, $t = 2.0$ from top to bottom. 45

Figure 4.4. The error comparison between polynomial fit differentiation, cubic spline differentiation and regular finite differences for disappearing and reappearing Gaussian Pulse. 46

Figure 4.5. The solution(a,c,e) and the adaptive grid(b,d,f) for the Poisson Equation at $t = 0$, $t = 0.25$, $t = 0.5$ from top to bottom. 47

Figure 4.6. Error vs. node number for two kinds of the WOFD and regular finite differences. The line marked with $\circ$ is for cubic splines, the line marked with $\square$ is for polynomial fits and the line marked with $\Diamond$ is for regular grid finite differences. 48

Figure 4.7. Error vs. time for two kinds of the WOFD for $\epsilon = 10^{-4}$ and $\epsilon = 10^{-5}$. – = cubic splines with $\epsilon = 10^{-4}$, - - = is cubic splines with $\epsilon = 10^{-5}$, the -$\circ$- = polynomial fit with $\epsilon = 10^{-4}$ and -·- = polynomial fit with $\epsilon = 10^{-5}$. 49

Figure 4.8. Number of active nodes vs. time for two kinds of the WOFD for $\epsilon = 10^{-4}$ and $\epsilon = 10^{-5}$. – = polynomial fit with $\epsilon = 10^{-5}$, - - = is cubic splines with $\epsilon = 10^{-5}$, -$\square$- = cubic spline with $\epsilon = 10^{-4}$ and -·- = polynomial fit with $\epsilon = 10^{-4}$. 50

Figure 4.9. The error vs. time with different stencil sizes for Poisson Equation. – = polynomial fit with 3 point stencil, - - = polynomial fit with 5 point stencil, -·- = cubic spline with 5 point stencil, -○- = cubic spline with 7 point stencil. 51

Figure 4.10. The solution(a,c,e) and the adaptive grid(b,d,f) for the Helmholtz Equation at $t = 0$, $t = 0.25$, $t = 0.5$ from top to bottom. 53

Figure 4.11. The error of the solutions vs. time with different Δt for Helmholtz Equation. – = time-step $\Delta t = 10^{-3}$, - - = time-step $\Delta t = 2.5\times10^{-3}$, -·- = time-step $\Delta t = 5 \times 10^{-3}$, -○- time-step $\Delta t = 10^{-2}$. 54

Figure 4.12. The error vs. node numbers. The line marked with ○ is for cubic splines, the line marked with □ is for polynomial fits and the line marked with ◊ is for regular grid finite differences. 55

Figure 4.13. The error vs. time with different ϵ. – = cubic splines with $\epsilon = 10^{-4}$, - - = cubic splines with $\epsilon = 10^{-5}$, -○- = polynomial fit with $\epsilon = 10^{-4}$, -·- = polynomial fit with $\epsilon = 10^{-5}$. 55

Figure 4.14. The error vs. time with different stencil sizes for Helmholtz Equation. – = polynomial fit with 3 point stencil, - - = polynomial fit with 5 point stencil, -○- = cubic spline with 5 point stencil, -·- = cubic spline with 7 point stencil. 56

Figure 4.15. The u vs. y profile on $x = 0.5$ (a) and the v vs. x profile on $y = 0.5$ (b) for steady-state $Re = 100$ and comparison to [1]. 57

Figure 4.16. The u vs. y profile on $x = 0.5$ (a) and the v vs. x profile on $y = 0.5$ (b) for steady-state $Re = 400$ and comparison to [1]. 58

Figure 4.17. The streamline patterns and the adaptive grid for steady state for $Re = 100$ (a) and $Re = 400$ (b). 59

Figure 4.18. The active node number vs. time for $Re = 100$ (a) and $Re = 400$ (b). 60

LIST OF SYMBOLS

$\vec{1}$	Vector of 1s
A	Discretized Poisson-Neumann Equation Matrix
a	A Parameter of Poisson or Helmholtz Problem
a_i	Constant Coefficient of i^{th} member Cubic Spline
b	A Parameter of Poisson or Helmholtz Problem
b_i	Coefficient of First Order Term of i^{th} member Cubic Spline
$\vec{b}$	Discretized Poisson-Neumann Equation Right Hand Side Vector
c	A Parameter of Poisson or Helmholtz Problem
c_i	Coefficient of Second Order Term of i^{th} member Cubic Spline
c_k^j	Coefficient of 1-D Basis Function of level j, index k
$c_{k,l}^j$	Coefficient of 2-D Basis Function of level j, index set k, l
d	A Parameter of Poisson or Helmholtz Problem
d_i	Coefficient of Third Order Term of i^{th} member Cubic Spline
d_k^j	Coefficient of 1-D Wavelet of level j, index k
$d_{k,l}^{j,\mu}$	Coefficient of 2-D Wavelet of level j, type μ, index set k, l
g	A Parameter of Poisson or Helmholtz Problem
h_i	Space between Two Consecutive Grid Points, i and $i+1$
Hr	Parameter of the Adjacent Zone to Sides
j	Level of Wavelet or Basis Location
j_0	Coarsest Level of Wavelets
J	Finest Level of Wavelets
$\vec{l}$	Left Null Space Vector of Discretized Poisson-Neumann Equation Matrix
N	Number of Total Time-steps
$\vec{n}$	Normal Vector to the Boundary
$p(x)$	Polynomial Fit for Spatial Discretization
p_i	Coefficients of Polynomial Fit for Spatial Discretization

$\vec{r}$	Right Null Space Vector of Discretized Poisson-Neumann Equation Matrix
r	Normalized Euclidian Norm
Re	Reynolds Number
Sl^n	n Point Stencil on 1-D
$S(x)$	Complete Cubic Spline
$S_i(x)$	i^{th} Member of Complete Cubic Spline
$\vec{u}$	Velocity Vector
$\vec{u^*}$	Non-Solenoidal Velocity Vector in Split-Step Method
$\vec{u_n}$	Velocity Vector at Time-step n
u	Velocity in x-dimension
v	Velocity in y-dimension
V_j	Set of Grid Points for the Basis Function Locations at Level j
Vr	Parameter of the Adjacent Zone to Upper and Lower j
W_j	Set of Grid Points for the Wavelet Locations at Level j
$w^j_{k,l}$	Coefficient of Polynomial Fit for a Wavelet
$\tilde{w}^j_{k,l}$	Coefficient of Polynomial Fit for a Basis Function
$\vec{x}$	Discretized Poisson-Neumann Equation Unknown Vector
Z^j	j^{th} Level of Set of Indices for Wavelets and Basis Functions
$\partial\Omega$	Boundary of the Solution Domain
α	A parameter of Dirichlet Boundary Condition
γ	Matrix that Holds Interpolation Values
Δt	Magnitude of the Time-step
ϵ	Wavelet Threshold
ε_i	Maximum Error at Time-step i
Φ	Correction Parameter for Navier-Stokes time-scheme
ϕ^j_k	1-D Basis Function of level j, index k
$\phi^j_{k,l}$	2-D Basis Function of level j, index set k, l
φ	Matrix that Holds Wavelet Values
σ	Second Derivative of Complete Cubic Spline

σ_i	Second Derivative of i^{th} Member of Cubic Spline
ψ_k^j	1-D Wavelet of level j, type μ, index k
$\psi_{k,l}^{j,\mu}$	2-D Wavelet of level j, type μ, index set k, l
Ω	Solution Domain

LIST OF ACRONYMS/ABBREVIATIONS

1-D	One Dimensional
2-D	Two Dimensional
CFD	Computational Fluid Dynamics
DNS	Direct Numerical Simulation
GMRES	Generalized Minimal Residual
ILU(n)	Incomplete LU Factorization of n Level of Fill
LES	Large Eddy Simulation
MPI	Message Passing Interface
PETSc	Portable Extensible Toolkit for Scientific Computations
PDE	Partial Differential Equation
SCALES	Stochastic Coherent Adaptive Large Eddy Simulation
WOFD	Wavelet Optimized Finite Differences

1. INTRODUCTION

Wavelets are compact functions in space and time that support multi-resolution analysis. They are efficient tools for data compression. With wavelets, any given regularly sampled data can be reconstructed and represented by a smaller number of nodes than the regular sampling rate. This allows them to be used in signal processing and image compression fields. They are also used for grid adaptation purpose by Computational Fluid Dynamics (CFD) community to solve Partial Differential Equations (PDEs), given their ability to compress data.

The grid adaptation, or data compression, comes naturally in wavelet methods. The idea is to generate a nested, multi-resolution grid. Then, each grid point is associated with a wavelet that supposedly lives on that point. Following the general principles of multiresolution analysis, wavelets that correspond to finer levels of the grid have more compact structures, i.e. smaller spatial region of influence. Each wavelet's multiplying coefficient is calculated. Then, the interpolation is represented by the superposition of these wavelets and their respective coefficients. The wavelets with coefficients smaller than an a-priori designated value and their corresponding grid points are discarded, and the irregular grid is obtained.

Once the adaptive grid is obtained, there are different ways to calculate the spatial derivatives of the interpolate. Three main ways are to use a Galerkin method, a collocation method or a finite difference or volume discretization. Wavelet Galerkin method requires wavelets to form a linearly independent basis, so the choice of wavelet type must be compatible. The wavelet collocation method uses the wavelets' known properties, such as an individual wavelet's derivative value at a given point. The finite difference or volume discretizations use wavelet compression only to obtain the irregular grid. These discretization methods calculate the spatial derivatives independent of the wavelet functions' properties.

Vasilyev *et al.* developed a wavelet collocation method [2]. They use the wavelets' known function properties to evaluate the spatial derivatives. Vasilyev and Paolucci later turned the method into an adaptive grid method [3], and presented results for multi dimensional problems [4]. All of these works used Daubechies' Wavelets' Autocorrelation function as a base wavelet.

Wavelet collocation method is also applied using interpolating wavelets by Vasilyev and Bowman [5]. Interpolating wavelets are also called second-generation wavelets. They support a faster wavelet transformation than Daubechies' wavelets. However, interpolating wavelets do not support the spatial differentiation procedure used in [3]. So, a new method for spatial differentiation is proposed based on the multilevel wavelet interpolation. This method is first applied to 1-D transient problems. The method is then generalized to multi dimensional transient equations by Vasilyev [6].

Wavelet collocation method using interpolating wavelets has been applied to several problems. One of these problems is the vorticity evolution equation to solve incompressible Navier-Stokes Equations with Brinkman Penalization to simulate stationary objects within the flow field, by Vasilyev and Kevlahan [7]. This work showed the method's ability to adapt the grid to vortex shedding behind a circular cylinder. The method was applied to high Reynolds Number flows, both with and without fluid structure interaction using a direct numerical simulation (DNS) like technique by Kevlahan and Vasilyev [8]. This paper [8] also introduced the notion to solve the incompressible Navier-Stokes Equations by a split-step (projection-correction) method with wavelet collocation. A new method is offered by Goldstein and Vasilyev [9], Stochasic Coherent Adaptive Large Eddy Simulation (SCALES), the Large Eddy Simulation (LES) equivalent of wavelets. Another application was to shock computations by Regele and Vasilyev [10]. All of these examples showed the easily applicable adaptive grid property of interpolating wavelets.

Jameson [11] is first to suggest to use wavelets for only grid adaptation, and do the differentiation by finite differences, calling it Wavelet Optimized Finite Differences (WOFD). He claims that using Galerkin method with wavelets makes it more

difficult than it needs to be, referencing the issue of dealing with nonlinearities and nonperiodic boundary conditions. He uses the knowledge coming from the wavelet multi-resolution property to calculate the spatial derivatives, and creates derivative calculation matrices. However, he does not use the derivatives of wavelets as functions directly. Using Daubechies' wavelets, he applies his notion to 1-D time dependent problems and demonstrates the method's potential.

Jameson updates his work from [11] in [12]. He suggests using algebraic, exponential or trigonometric polynomial fits for the spatial derivative calculations. He gives some details on how to generate the differentiation coefficients using algebraic polynomial fits on the irregular data. As an alternative, Walden suggests to use convolution operators directly on the irregular grid as differentiation tools in his work [13]. However this method is much more complex than Jameson [12]. Jameson also proposes techniques to use adaptive differentiation order on the adaptive grid, i.e. the finite difference order changes within the domain. He applies this technique also on 1-D time-dependent problems, claiming the possibility to use it for 2-D.

Holmstrom suggests a different finite difference method using wavelet adapted grid [14]. His idea is to interpolate ghost points around the active wavelet to form a regular finite difference stencil. Then he suggests to do the differentiation on this regular finite difference grid. This idea has the benefit of being possibly more accurate than working directly on the irregular grid, but the process of activating ghost points is cumbersome.

Jameson and Miyama describe WOFD on 2-D combining Daubechies' wavelets and polynomial fit based differentiation on the irregular grid in their work [15]. They present their method on an oceanography problem. The wavelet adaptive grid seems to work, and some implementation notions about adaptive grid stability is given, such as the need to use centered stencils. This method is called WOFD-Adaptive High Order. In fact the order of error is 4, but this is claimed to be a high order of error for oceanography.

Cruz *et al*. utilize interpolating wavelets for WOFD analysis, using cubic splines instead of polynomial fits on the irregular grids [16]. They apply this method to 1-D transient problems. Then, Santos *et al*. generalize this method to multi-dimensional transient problems with applications to chemical engineering problems, but not the incompressible Navier-Stokes Equations [17]. They do not comment in detail on finite difference grid stability, but the adaptive grids they obtained have the signs of some extra measures in this regard.

Yousefi and Noorzad use WOFD as a combination of interpolating wavelets and polynomial fit based differentiation [18]. They apply this method to a wave propagation problem in a solid medium. They do not comment significantly on either the grid stability, or the spatial differentiation process.

In this thesis, interpolating wavelets and two different finite difference implementations, polynomial fit based and cubic spline based, are combined. Some suggestions are made for certain implementation issues, most importantly the grid stability. Comparisons between two spatial differentiation variants are made. The resulting solver is applied to split-step method for transient incompressible Navier-Stokes Equations.

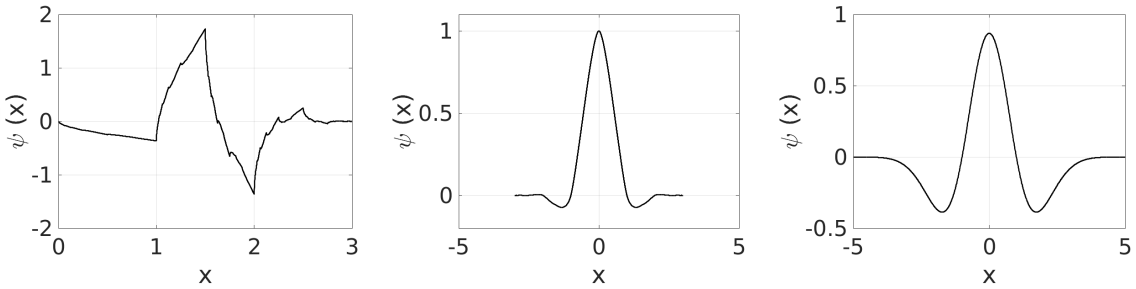
The split-step (projection-correction) method is preferred over stream function-vorticity and vorticity-velocity formulations, even though it is harder to implement. This is because that the vorticity-velocity function is reported to show high errors for Brinkman Penalization method for high Reynolds Number flows [8]. Also, the main idea is to demonstrate the WOFD's applicability to the split-step method, rather than finding the solutions to well-established benchmark problems.

Portable Extensible Toolkit for Scientific Computation (PETSc) is used as a framework in the application of WOFD. It is a set of C-language libraries that supports parallelism through Message Passing Interface (MPI). It provides high level tools for vector and matrix calculations and linear and non-linear solvers. Several parallel sparse matrix representations and linear and non-linear solver types are supported, which makes PETSc a great tool for scientific parallel computing.

The thesis is constructed as follows. First, 1-D wavelets and the differentiation procedures are explained. Then, 2-D wavelets are described and the implementation of a stable irregular grid is given. After that, the model problems and the Navier-Stokes Equations are defined. The solution for these problems are given. Finally, the thesis is finalized with concluding remarks.

2. WAVELET ANALYSIS

Wavelet analysis may be performed with different kinds of wavelets. Three of these kinds are represented in Figure 2.1. The spatial compactness of the wavelets can be seen clearly.



(a) (b) (c)

Figure 2.1. Different kinds of wavelets, (a) Daubechies' 2 Wavelet, (b) Autocorrelation of Daubechies' 2, (c) Mexican Hat Wavelet.

Partial differential equation (PDE) applications of wavelet analysis may be performed through the implementation of either one of the three basic approaches: Wavelet Galerkin, Wavelet Collocation or Wavelet Optimization. Wavelet Galerkin and Collocation methods use properties of wavelet functions directly, whereas wavelet optimized methods perform their independent spatial differentiations using wavelets only for irregular grid construction.

Wavelet Galerkin methods use Galerkin method, hence the name, to calculate the stiffness matrix of a partial differential equation (PDE). These methods in general transform the interpolate into wavelet function space, solve the problem there, and transform the solution back to the physical domain. This results in difficulties when dealing with non-linear terms, such as those of the Navier-Stokes equations, or with non-periodic boundary conditions.

Wavelet Collocation methods use the wavelet's function and interpolation properties to calculate the derivatives, but does so in the physical domain. Remaining in

the physical domain avoids the above difficulties encountered in the Wavelet Galerkin method. To calculate the derivative at a given point, either the contribution by all the wavelets that can affect that point is summed up, or the known properties of the overall wavelet interpolation is exploited. In the case of interpolating wavelets, known property of the overall interpolation is that the interpolation coincides with a polynomial fit on each wavelet level.

Wavelet optimized methods use wavelets only for grid creation. Once the irregular grid is constructed, these methods use their independent finite differences, volume, or element discretization. To use this, there are two main choices. Either the approximation is interpolated onto a regular fine domain and the finite differences or volume is applied, or the finite differences or volume is applied directly on the irregular grid. In this work, this approach is used with finite differences being applied directly on the irregular grid.

Applying finite differences directly on the irregular grid is possible by creating a local analytical approximation on the stencil. To create this analytical approximation two approaches are possible, applying polynomial fits or applying cubic splines. Both approaches are used in this work and they are compared with each other.

2.1. 1-D Multi-Resolution Wavelet Analysis

To understand the basic principle of wavelet analysis, first 1-D models are studied. Wavelets are constructed on a nested, multi-resolution grid. Say that a grid of level J is the union of two grids, the grid of one level below and its complementary on the level J. This proposition can be seen in equation 2.1

$$V_J = V_{J-1} \cup W_{J-1} \tag{2.1}$$

where V_J is the basis grid on the level J, V_{J-1} is the basis grid on the level $J-1$ and W_{J-1} is the complementary of V_{J-1} on level J. This deconstruction can be performed

until a wanted coarsest level of j_0 is reached.

$$V_J = V_{j_0} \cup W_{j_0} \cup W_{j_0+1} .... \cup W_{J-2} \cup W_{J-1} \tag{2.2}$$

The idea of this deconstruction can also be seen in Figure 2.2. The parameters are $J = 6$ and $j_0 = 2$. The level $j_0 - 1$ represents the basis function locations.

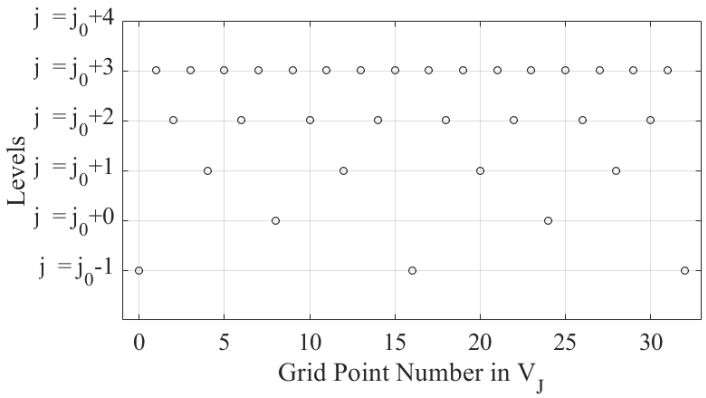


Figure 2.2. The nested grid decomposed to its level for 1-D

Every point of every W_j is associated with a wavelet $\psi_m^j(x)$ and every point in the basis grid V_{j_0} is associated with a basis function $\phi_m^{j_0}(x)$. In the light of this nested multi-resolution grid, any given regularly sampled data of a function can be thought of as given basis function coefficients at a given level J. This can be mathematically shown as equation 2.3

$$f(x) = \sum_{k \in Z^J} c_k^J \phi_k^J(x) \tag{2.3}$$

where k is the identifying number of the basis functions and J is the level in which these basis functions live.

Using the coefficients in these equations, one lower level can be approximated by a combination of basis and wavelets coefficients as can be seen in the equation 2.4. The calculation of these coefficients is performed using the lifting scheme as summarized in the paper of Vasilyev and Bowman [5]. This scheme is explained in the upcoming first

subheading.

$$\sum_{k \in Z^J} c_k^J \phi_k^J(x) = \sum_{k \in Z^{J-1}} c_k^{J-1} \phi_k^{J-1}(x) + \sum_{m \in Z^{J-1}} d_m^{J-1} \psi_m^{J-1}(x) \tag{2.4}$$

In equation 2.4, $\sum_{k \in Z^{J-1}} c_k^{J-1} \phi_k^{J-1}(x)$ can also be split into lower level wavelet and basis function coefficients by the lifting scheme as can be seen in equation 2.5.

$$\sum_{k \in Z^{J-1}} c_k^{J-1} \phi_k^{J-1}(x) = \sum_{k \in Z^{J-2}} c_k^{J-2} \phi_k^{J-2}(x) + \sum_{m \in Z^{J-2}} d_m^{J-2} \psi_m^{J-2}(x) \tag{2.5}$$

This process can be done until a desired level of coarseness of j_0 to get the initial interpolation function as a combination of basis function coefficients at j_0 and wavelet function coefficients from $J-1$ to j_0 as can be seen in equation 2.6.

$$f(x) = \sum_{k \in Z^{j_0}} c_k^{j_0} \phi_k^{j_0}(x) + \sum_{j=j_0}^{J-1} \sum_{m \in Z^j} d_m^j \psi_m^j(x) \tag{2.6}$$

2.1.1. 1-D Lifting Scheme

Wavelet representation of an interpolation function is done using 2^{nd} order wavelets (interpolating wavelets) lifting scheme. This scheme is facilitated based on the work of Vasilyev and Bowman [5]. Forward wavelet transformation by this scheme can be summarized as equations 2.7 and 2.8.

$$d_k^j = \frac{1}{2}(c_{2k+1}^{j+1} - \sum_l w_{k,l}^{j+1} c_{2k+2l}^{j+1}) \tag{2.7}$$

$$c_k^j = c_{2k}^{j+1} + \sum_l \tilde{w}_{k,l}^j d_{k+l}^j \tag{2.8}$$

Backwards wavelet transform is given by equations 2.9 and 2.10 . These simply take back the steps provided by equations 2.7 and 2.8.

$$c_{2k}^{j+1} = c_k^j - \sum_l \tilde{w}_{k,l}^j d_{k+l}^j \tag{2.9}$$

$$c_{2k+1}^{j+1} = 2d_k^j + \sum_l w_{k,l}^{j+1} c_{2k+2l}^{j+1} \tag{2.10}$$

In the previous equations, $\tilde{w}_{k,l}^j$ and $w_{k,l}^{j+1}$ appear to represent the polynomial fit given by the basis or wavelet coefficients. In equation 2.7, the rough idea is to have a polynomial fit using a portion of the basis functions, and then setting the wavelet coefficient as the difference between the polynomial fit and the actual value at that location. Same principle appears at equation 2.8, where the polynomial fit is now given by the wavelet functions in order to provide "the lifting scheme". Note that the orders of the polynomials does not have to be same for equations 2.7 and 2.8, so the weights are represented with different notations(one with a tilde). Polynomial fit of order 5 is used in this work for both cases, i.e. 6 coefficients were used for each $\sum$.

2.1.2. 1-D Compression of Interpolation Function

The strength of the wavelet transformation is that it allows easy grid adaptation. The wavelet coefficients whose magnitudes are less than a chosen ϵ value in equation 2.6 can be omitted from the expression, and still a good approximation can be obtained. In mathematical terms, this notion can be summarized as equation 2.11.

$$f(x) \approx \sum_{k \in Z^{j_0}} c_k^{j_0} \phi_k^{j_0}(x) + \sum_{j=j_0}^{J-1} \sum_{\substack{m \in Z^j \\ |d_m^j| \geq \epsilon}} d_m^j \psi_m^j(x) \tag{2.11}$$

The compressed function allows the user to find the irregular grid. It is known that the wavelet coefficients are below an ϵ value where the gradient of the interpolated function is low enough. They become significant at the high gradient locations. This describes how to use the compressed function to generate the irregular grid. Keep the grid points associated with the wavelets of coefficients greater than epsilon, discard the others.

Also, wavelet locations which may become significant in the future have to be kept, even though their coefficients are below ϵ. This allows tracking the high gradient regions in time. This is done by turning on certain neighbouring wavelets. The mathematical expression for a neighbouring wavelet $\psi_{k'}^{j'}(x)$ for any wavelet $\psi_k^j(x)$ is given by equation 2.12

$$|j' \pm Vr| \geq j \text{ and } Hr \geq |2^{j'-j}k - k'| \tag{2.12}$$

where Vr tells how many levels coarser or denser to go and Hr tells how many wavelets to the sides to go.

2.1.3. Implementation of 1-D Wavelet Transform

Implementation of the wavelet transform is done to use as little memory as possible. Wavelet and basis coefficients are kept on a single $n \times 1$ matrix, the sampled values of the function are kept on a single $n \times 1$ matrix and the adaptive grid is kept on an $n \times 1$ matrix . All matrices are in sparse matrix format to reduce the memory use. The foundation of this idea is given by Wirasaet [19]. PETSC package is used for the implementation [20, 21].

The code is written such that at $j_0 = 4$ level of resolution there are 8 wavelets and 9 basis functions, in total 17 grid points.

2.1.3.1. Regular Grid. On a regular grid, with the highest level of resolution J, the number of grid points is given by the equation,

$$n = 2^{J-1} + 1 \tag{2.13}$$

Say that the matrix which holds the values of the function is called γ, the values which will hold the values of the wavelet and basis coefficients is called φ. They are both matrices of size $n \times 1$.

For a given J and j, the algorithm to find the wavelet coefficients is given as algorithm given in Figure 2.3. Remember that J is defined as the finest level of resolution with the sampled data as in equation 2.3.

$NumberOfWavelets = 2^{j-1}$;
for $i = 0$ to $NumberOfWavelets$ **do**
  $m = i2^{J-j} + 2^{J-j-1}$;
  $\varphi(m, 1) = \frac{1}{2}(\varphi(m, 1) - \sum_l w_l \varphi((i + l)2^{J-j}, 1))$;
end for

Figure 2.3. Forward wavelet coefficient calculation.

Following the same approach, the basis function coefficients are found as algorithm given in Figure 2.4.

$NumberOfBasis = 2^{j-1} + 1$;
for $i = 0$ to $NumberOfBasis$ **do**
  $m = i2^{J-j}$;
  $\varphi(m, 1) = \varphi(m, 1) + \sum_l w_l \varphi((i + l)2^{J-j} + 2^{J-j-1}, 1))$;
end for

Figure 2.4. Forward basis function coefficient calculation.

Using algorithms given in Figures 2.3 and 2.4, the overall wavelet transformation from level J to level j_0 can be constructed as algorithm given in Figure 2.5.

Copy γ onto φ;
for $j = J - 1$ to j_0 **do**
  $NumberOfWavelets = 2^{j-1}$;
  for $i = 0$ until $i \geq NumberOfWavelets$ **do**
    $m = i2^{J-j} + 2^{J-j-1}$;
    $\varphi(m, 1) = \frac{1}{2}(\varphi(m, 1) - \sum_l w_l \varphi((i + l)2^{J-j}, 1))$;
  end for
  $NumberOfBasis = 2^{j-1} + 1$;
  for $i = 0$ until $i \geq NumberOfBasis$ **do**
    $m = i2^{J-j}$;
    $\varphi(m, 1) = \varphi(m, 1) + \sum_l w_l \varphi((i + l)2^{J-j} + 2^{J-j-1}, 1))$;
  end for
end for

Figure 2.5. Forward wavelet transformation.

2.1.3.2. Irregular Grid. Suppose that an irregular grid data is provided with a boolean matrix *grid*. This matrix has the size of a regular grid $n \times 1$. In this matrix, the value is 1 if the corresponding grid location is on and the value is absent according to the PETSc's sparse matrix representation format if the corresponding grid location is off.

Using the same idea, the matrix that holds the approximations of regular grid points, γ and the matrix that holds the wavelet coefficients, φ, have both $n \times 1$ size. But, they have values only if the corresponding grid point is on, otherwise the spots on the matrices are absent according to the sparse matrix representation.

Following the previous points, the forward wavelet transformation can be written as algorithm given in Figure 2.6.

Copy γ onto φ;
for $j = J - 1$ to j_0 **do**
  $NumberOfWavelets = 2^{j-1}$;
  for $i = 0$ until $i \geq NumberOfWavelets$ **do**
    $m = i2^{J-j} + 2^{J-j-1}$;
    if $grid(m, 1) = 1$ **then**
      $\varphi(m, 1) = \frac{1}{2}(\varphi(m, 1) - \sum_l w_l \varphi((i + l)2^{J-j}, 1))$;
    end if
  end for
  $NumberOfBasis = 2^{j-1} + 1$;
  for $i = 0$ until $i \geq NumberOfBasis$ **do**
    $m = i2^{J-j}$;
    if $grid(m, 1) = 1$ **then**
      $\varphi(m, 1) = \varphi(m, 1) + \sum_l w_l \varphi((i + l)2^{J-j} + 2^{J-j-1}, 1))$;
    end if
  end for
end for

Figure 2.6. Forward wavelet transformation on an irregular grid.

2.2. Spatial Differentiation on Irregular Grid in 1-D

The two ways of creating the finite difference scheme to calculate the spatial derivatives are explained here. These are mainly the polynomial fit approach, and the cubic spline approach. Polynomial fit approach suggests to fit a polynomial on the irregular grid, whereas the cubic spline approach uses cubic splines to fit the data. The second step is the same for both approaches, the derivative is calculated analytically on the given polynomial fit or cubic spline.

2.2.1. Polynomial Fit Approach

The only point of the wavelet transformation is to get an irregularly spaced grid points. The derivative calculation does not use wavelet techniques in this work. One of the ways to calculate the derivatives of the interpolation function $f(x)$ on grid points is the algebraic polynomial approximation scheme.

Assume that we have an n point stencil, S^n. Suppose that we want to calculate d^{th} derivative on the m^{th} point. To do this, we put a polynomial fit on the stencil points. Also assume that the function can be approximated using a polynomial fit $p(x_i) = f_i$ with given function values on the domain $f(x_i) = f_i$. This assumption is fair since we use interpolatory wavelets, and the characteristic of these wavelets is that they approximate polynomial fits.

The definition of the polynomial is as equation 2.14,

$$p(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + ... + c_{n-1} x^{n-1} \tag{2.14}$$

This polynomial can then be differentiated analytically as equations 2.15,

$$\begin{aligned} p'(x) &= c_1 + 2c_2 x^1 + 3c_3 x^2 + ... + (n-1)c_{n-1} x^{n-2} \\ p''(x) &= (2 \cdot 1)c_2 + (3 \cdot 2)x^3 + ... + (n-1)(n-2)c_{n-1} x^{n-3} \end{aligned} \tag{2.15}$$

The new question arises here on what the coefficients c_i are. Knowing f_i and that $p(x)$ must satisfy $p(x_i) = f_i$, we can write a linear system of equations to solve for c_i as equation 2.16.

$$c_0 + c_1 x_i + c_2 x_i^2 + c_3 x_i^3 + ... + c_{n-1} x_i^{n-1} = f_i \ , \ x_i \in S^n \tag{2.16}$$

Alternatively, this system of equations can be written in dense matrix representation as equation 2.17.

$$\begin{bmatrix} 1 & x_1 & x_1^2 & . & . & x_1^{n-1} \\ 1 & x_2 & x_2^2 & . & . & x_2^{n-1} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1 & x_n & x_n^2 & . & . & x_n^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ . \\ . \\ . \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ . \\ . \\ . \\ f_n \end{bmatrix} \tag{2.17}$$

This system is solved for $n = 7$ in this study. So, given that it is a sufficiently small system, a direct matrix inversion in PETSc's low-level kernel is used in its solution.

After this point, having the coefficients c_i, the derivative of this polynomial is available for all x. For accuracy purposes, we always choose the derivation location from the stencil points.

2.2.2. Cubic Spline Approach

A different way to obtain the spatial derivative in an irregular grid is to form a cubic spline along the stencil points. Suppose we have an $n + 1$ point stencil with

coordinates $\{x_0, x_1, x_2, x_3, ..., x_n\}$. A cubic spline is defined such that

$$S(x) = S_i(x) \text{ at } [x_i x_{i+1}] \text{ for } i = 0, 1, .., n-1 \tag{2.18}$$

where

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{2.19}$$

This means that any two neighbouring points are connected with a cubic polynomial. There are additional criteria to have unique solutions to the coefficients of the polynomials as equation 2.20 .

$$\begin{aligned} S_i(x_i) &= f_i \\ S_{n-1}(x_n) &= f_n \\ S_i(x_{i+1}) &= S_{i+1}(x_{i+1}) \\ S_i^{'}(x_{i+1}) &= S_{i+1}^{'}(x_{i+1}) \\ S_i^{''}(x_{i+1}) &= S_{i+1}^{''}(x_{i+1}) \end{aligned} \tag{2.20}$$

However, these conditions are not enough by themselves. Two more conditions are needed. In this paper, since the point of this analysis is to calculate the derivatives, two conditions are chosen as not-a-knot conditions. Not-a-knot conditions are as equation 2.21.

$$\begin{aligned} S_0^{'''}(x_1) &= S_1^{'''}(x_1) \\ S_{n-2}^{'''}(x_{n-1}) &= S_{n-1}^{'''}(x_{n-1}) \end{aligned} \tag{2.21}$$

Other conditions are possible, such as fixing second derivatives to an a-priori value. But these conditions could be harmful for the accuracy of the inteprolation, so

not-a-knot is chosen.

After working on the conditions, it can be seen that the following relationships between the coefficients and the second derivative of the spline at stencil points σ_i hold. Defining $h_i = x_{i+1} - x_i$,

$$\begin{aligned} a_i &= \frac{\sigma_{i+1} - \sigma_i}{6h_i} \\ b_i &= \frac{\sigma_i}{2} \\ c_i &= \frac{f_{i+1} - f_i}{h_i} - h_i \frac{2\sigma_i \sigma_{i+1}}{6} \\ d_i &= f_i \end{aligned} \tag{2.22}$$

After inserting these equalities in equation set 2.20, and applying not-a-knot conditions, the linear system is obtained as equations 2.23.

$$\begin{gathered} \sigma_0 \frac{-1}{h_0} + \sigma_1 (\frac{1}{h_0} + \frac{1}{h_1}) + \sigma_2 \frac{-1}{h_1} = 0 \\ \sigma_{i-1} \frac{h_{i-1}}{6} + \sigma_i (\frac{h_{i-1}}{3} + \frac{h_i}{3}) + \sigma_{i+1} \frac{h_i}{6} = \frac{f_{i-1} - f_i}{h_{i-1}} + \frac{f_{i+1} - f_i}{h_i} \\ \text{for } i = 1, ..., n-1 \\ \sigma_{n-2} \frac{-1}{h_{n-2}} + \sigma_{n-1} (\frac{1}{h_{n-2}} + \frac{1}{h_{n-1}}) + \sigma_n \frac{-1}{h_{n-1}} = 0 \end{gathered} \tag{2.23}$$

After solving this set of equations and putting the solution in equations 2.22, the analytical description for the cubic spline is obtained. Using the analytical description, the derivatives of the interpolation are calculated at stencil points.

2.3. 2-D Multi-Resolution Wavelet Analysis

The idea of creating a nested grid in 2-D is very similar to its counterpart in 1-D. The equations 2.1 and 2.2 for multi-resolution grid sets hold also for 2-D. A representation of a 2-D nested grid can be seen in Figure 2.7. As before, each grid point is assigned a wavelet or a basis function of the same level. In Figure 2.7, the square shapes represent the basis locations in level $j_0 = 2$, diamond shapes represent

the wavelet locations on level $j = 2$ and the dots represent the wavelet locations on level $j = 3$.

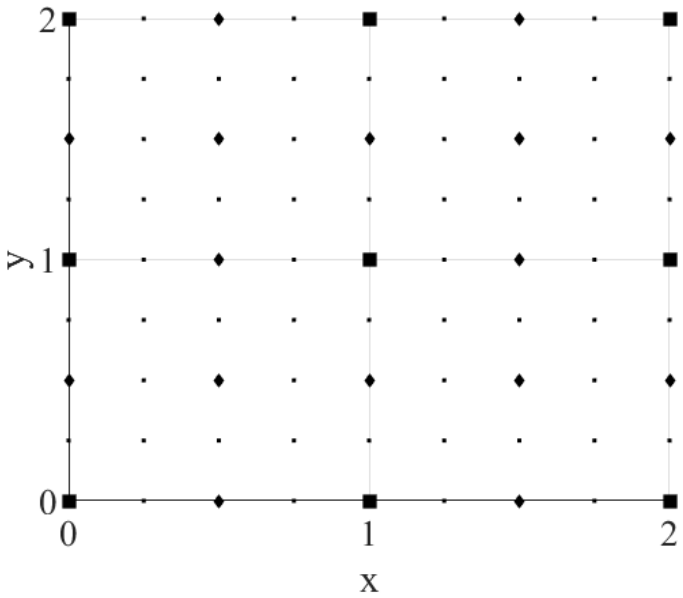


Figure 2.7. The nested grid decomposed to its levels for 2-D. ■ = Basis Function of level $j = 2$, ♦ = Wavelets of level $j = 2$, · = Wavelets of level $j = 3$.

There are two basic approaches to use a wavelet transform on 2-D, 2-D wavelets or dyadic wavelets that are created by multiplication of 1-D wavelets on 2-D. The second approach is used in this work, as it is simpler to implement once the implementation code for 1-D is written. Let a function depending on two space dimensions be written as a superposition of some appropriate basis functions as can be seen in equation 2.24.

$$f(x,y) = \sum_{k \in Z^J} \sum_{l \in Z^J} c^J_{k,l} \phi^J_{k,l}(x,y) \tag{2.24}$$

where $\phi^J_{k,l}(x,y)$ is the dyadic multiplication of two 1-D basis functions as can be seen in equation 2.25.

$$\phi^J_{k,l}(x,y) = \phi^J_k(x)\phi^J_l(y) \tag{2.25}$$

Taking one step on one level lower, the approximation function becomes equation 2.26.

$$f(x,y) = \sum_{k\in Z^{J-1}} \sum_{l\in Z^{J-1}} c_{k,l}^{J-1}\phi_{k,l}^{J-1}(x,y) + \sum_{\mu=1}^{3} \sum_{k\in Z^{J-1}} \sum_{l\in Z^{J-1}} d_{k,l}^{J-1,\mu}\psi_{k,l}^{J-1,\mu}(x,y) \tag{2.26}$$

where $\phi_{k,l}^{J-1}(x,y)$ and $\psi_{k,l}^{J-1,\mu}(x,y)$ are the dyadic multiplications as can be seen in equation set 2.27.

$$\begin{aligned}
\phi_{k,l}^{J-1}(x,y) &= \phi_k^{J-1}(x)\phi_l^{J-1}(y) \\
\psi_{k,l}^{J-1,1}(x,y) &= \phi_k^{J-1}(x)\psi_l^{J-1}(y) \\
\psi_{k,l}^{J-1,2}(x,y) &= \psi_k^{J-1}(x)\phi_l^{J-1}(y) \\
\psi_{k,l}^{J-1,3}(x,y) &= \psi_k^{J-1}(x)\psi_l^{J-1}(y)
\end{aligned} \tag{2.27}$$

A new notion that does not appear in 1-D analysis occurs here, called wavelet type, denoted with μ. It helps in distinguishing between two wavelets which otherwise have same subscripts, k, l, but associated with different grid points on the domain being generated by different dyadic multiplications.

By applying this step on the newly generated basis functions until a level of wanted coarseness, j_0 is reached, the function's wavelet interpolation is concluded as equation 2.28.

$$f(x,y) = \sum_{k\in Z^{j_0}} \sum_{l\in Z^{j_0}} c_{k,l}^{j_0}\phi_{k,l}^{j_0}(x,y) + \sum_{j=j_0}^{J-1} \sum_{\mu=1}^{3} \sum_{k\in Z^{j}} \sum_{l\in Z^{j}} d_{k,l}^{j,\mu}\psi_{k,l}^{j,\mu}(x,y) \tag{2.28}$$

2.3.1. 2-D Lifting Scheme

Lifting scheme on 2-D is nothing but the consecutive application of 1-D Lifting scheme on two directions, x and y. Since the 2-D wavelets are the dyadic multiplications of their 1-D counterparts, this approach yields correct results.

To take the forward wavelet transform, for a given level of coarseness j, apply lifting scheme equations 2.7 and 2.8 along the x direction, on all y coordinates that contain $\phi^j_{k,l}(x,y)$, $\psi^{j,1}_{k,l}(x,y)$, $\psi^{j,2}_{k,l}(x,y)$ and $\psi^{j,3}_{k,l}(x,y)$.

Beware a pitfall here. Note that there may be coordinates that contain wavelets of higher orders, $j+i$, between aforementioned y coordinates. But level j of lifting must not be applied on any of these $j+i$ coordinates.

The completion of a one level wavelet transform with lifting scheme is to apply equations 2.7 and 2.8 this time on y direction on all x coordinates that contain $\phi^j_{k,l}(x,y)$, $\psi^{j,1}_{k,l}(x,y)$, $\psi^{j,2}_{k,l}(x,y)$ and $\psi^{j,3}_{k,l}(x,y)$. Beware the same pitfall as above.

2.3.2. 2-D Compression of Interpolation Function

The 2-D compression of the wavelet transformation lies on the same idea with its 1-D counterpart. If there is a sharp gradient, the finer level wavelets attain significant coefficients. If not, the magnitudes of their coefficients are small and can be neglected, leading us to equation 2.29. Neglecting wavelets with coefficients of magnitude smaller than ϵ does not significantly cripple the interpolation.

$$f(x,y) \approx \sum_{k\in Z^{j_0}} \sum_{l\in Z^{j_0}} c^{j_0}_{k,l}\phi^{j_0}_{k,l}(x,y) + \sum_{j=j_0}^{J-1} \sum_{\mu=1}^{3} \sum_{\substack{k\in Z^j \\ |d_{k,l}|\geq\epsilon}} \sum_{\substack{l\in Z^j \\ |d_{k,l}|\geq\epsilon}} d^{j,\mu}_{k,l}\psi^{j,\mu}_{k,l}(x,y) \tag{2.29}$$

Note that the basis function locations are not omitted, similar to 1-D. This is done to keep the stability of the wavelet transform.

The adjacent zone creation is performed according to the guideline of equation 2.12. However, since we are now in 2-D, there are different paths to choose with this guideline. Recall that a 2-D wavelet $\psi^{j,\mu}_{k,l}(x,y)$ is the dyadic multiplication of two

1-D wavelets $\psi_{k_1}^{j_1}(x)$ and $\psi_{l_2}^{j_2}(y)$ (or one of these is a basis function, but for simplicity it is represented as a wavelet here). Depending on the greatness relation between j_1 and j_2, k,l and μ are assigned.

The first option is to generate the 1-D adjacent zones in each direction, using the level and number sets j_1, k_1 and j_2, k_2. For a case where $j_1 = 5$, $k_1 = 8$, $j_2 = 2$, $k_2 = 1$, $J = 6$, $j_0 = 2$,the application of this approach looks like Figure 2.8. The adjacent zone created by $\psi_{k_1}^{j_1}(x)$ is represented as dots and the adjacent zone of $\psi_{k_2}^{j_2}(x)$ is represented by crosses.

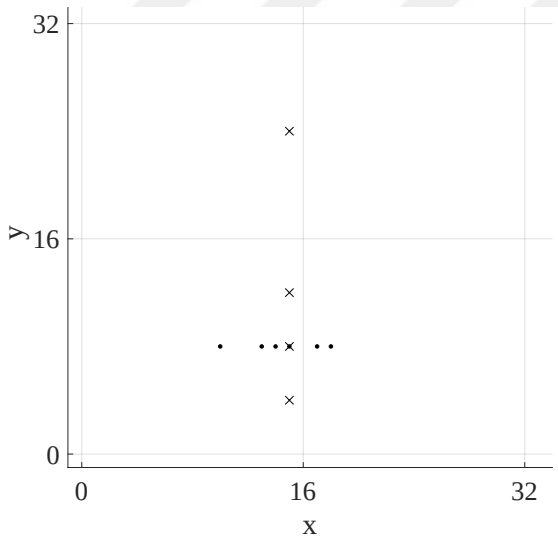


Figure 2.8. First option for adjacent zone creation in 2-D. $\cdot$ = adjacent zone in x, $\times$=adjacent zone in y.

As it can be seen in Figure 2.8, this is not a "true" 2-D approach. The corner wavelets are not considered in the adjacent zone, even though they are as related to the original wavelet as the wavelets that are included.

The second option is to generate a 2-D square adjacent zone from each dimension's 1-D adjacent zones. Each dimension's adjacent zone extends also to the other direction with the same distances from the original point. For the same case as Figure 2.8, this approach yields more satisfactory results as can be seen in Figure 2.9. Here again,

the adjacent zone of $\psi_{k_1}^{j_1}(x)$ is represented as dots and the adjacent zone of $\psi_{k_2}^{j_2}(x)$ is represented by crosses.

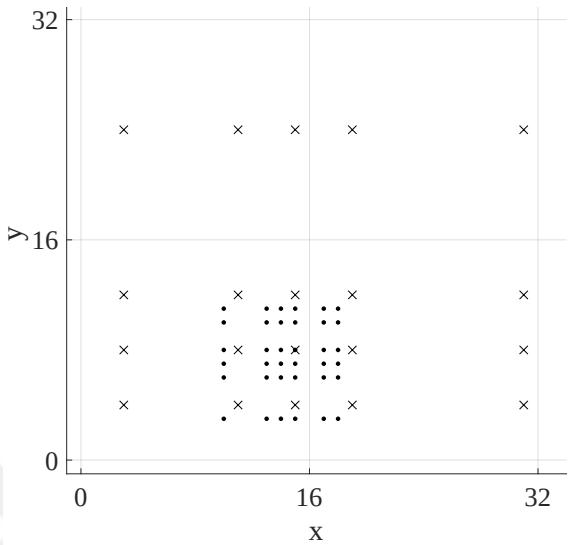
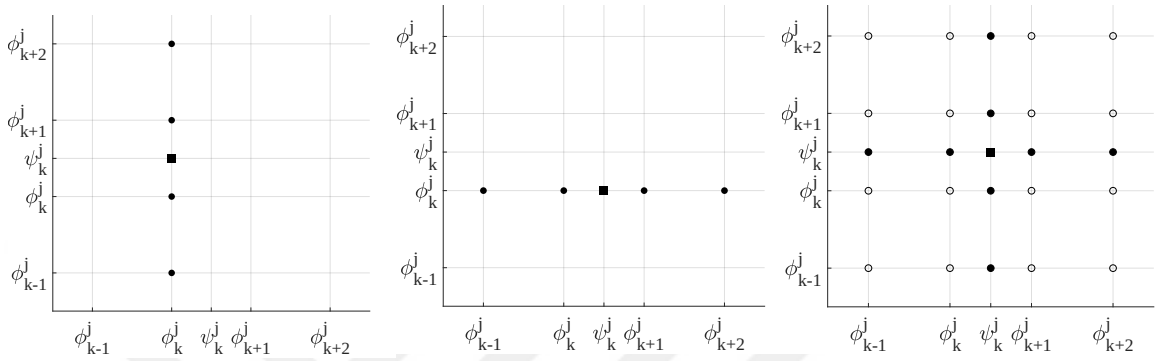


Figure 2.9. Second option for adjacent zone creation in 2-D. $\cdot$ = adjacent zone created by the wavelet on x, $\times$ = adjacent zone created by the wavelet on y.

In Figure 2.9, corner wavelets are also activated. There is a + shaped gap in the middle of the inner adjacent zone. In theory, this gap may seem problematic. However, in practice, with the addition of the precautions for the finite difference stencil stability, this gap does not cause any problems. Also, since there is almost never a case where a finer level wavelet is active with its coarser level neighbour wavelets inactive, that gap almost never exists in the first place. Notice that there is also more adjacent zone points than there should be according to the equation 2.12. The equation is not changed. This is due to the implementation of the equation in "int" data-type of C-language.

2.3.2.1. Minimum Set Construction. The wavelet decomposition depends on using several points to create a polynomial fit, and then taking the difference of the polynomial fit with the actual value at that location. For this reason, a minimum set of wavelet locations must be present for every active point on the domain for a wavelet transformation to be possible.

Since wavelet transformation is a level-by-level exercise, thinking this minimum set requirement in a level-by-level manner makes things easy. For each wavelet type, $\mu = 1, 2, 3$ the minimum set locations can be seen in Figure 2.10 for 3^{rd} order polynomial fits.



(a) (b) (c)

Figure 2.10. Minimum set points for three types of wavelets, (a) is for type 1, (b) is for type 2 and (c) is for type 3. ■ = the originally active grid point,○ and ● = the minimum set points.

Using the idea from Figure 2.10, the minimum set is constructed for every active wavelet point. Note that before applying the minimum set condition, the only active points will be $d^{J-1}_{k,l} > \epsilon$ for the finest level $J-1$. For the coarser levels, the active points before the application of the minimum set will be $d^{j}_{k,l} > \epsilon$ and the minimum set points coming from finer levels $j' > j$. The algorithm for this procedure can be seen in Figure 2.11.

Given an Irregular Grid G;
for $j = J-1$ to j_0 **do**
  for $\mu = 1$ to 3 **do**
    Activate Minimum Set Points of all active points of $\psi^j_{k,l}(x, y)$ in G;
  end for
end for

Figure 2.11. Minimum set construction.

Applying this algorithm creates the grid in Figure 2.12 for 3^{rd} order polynomials. In Figure 2.12 only one wavelet point is originally active, $d^{J-1}_{7,0}$. Wavelet parameters are $J = 6$, $j_0 = 3$.

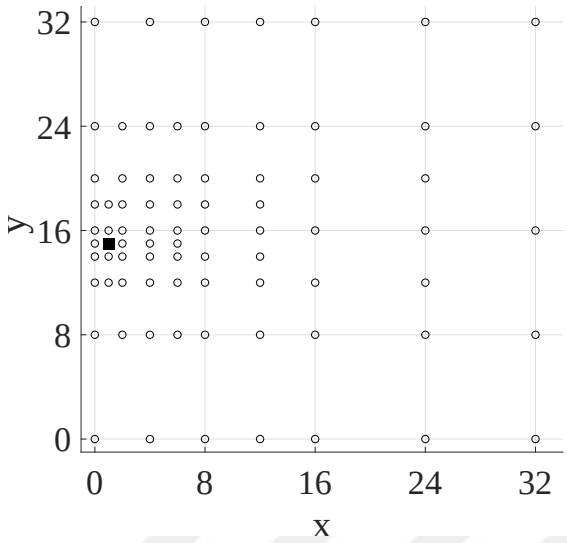


Figure 2.12. An example minimum set grid where the square is the originally active wavelet.

2.4. Spatial Differentiation on Irregular Grid in 2-D

Differentiation on 2-D is done using the 1-D approach. In each direction, x and y, a number of neighbouring points are chosen and used to create a polynomial fit or a cubic spline according to the second heading 2.2. A sample stencil can be seen in the Figure 2.13 for 4^{th} order polynomials.

In Figure 2.13, the point denoted with a circle is the point we would like to differentiate, and the points that are marked with squares are the respective stencil points.

As it can be seen, this approach requires active grid points in the main directions, namely positive and negative x and y directions. If the wavelet compression fails to provide every point with enough number of stencil points, this method is not applicable. In general, wavelet compression is used with minimum set construction, so this requirement is fulfilled.

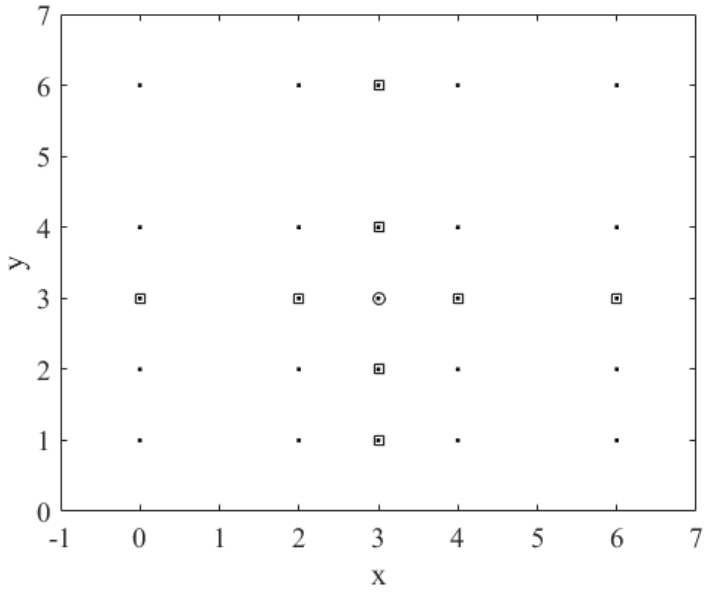


Figure 2.13. A centered stencil. Active points are denoted with ·, the differentiation point is marked with ∘ and the stencil points are marked with □ in addition to ·.

There is another, more subtle criterion for satisfactory performance of this approach. A point cannot be its stencil's edge point if it is not a boundary point for the entire solution domain. The reason for this criterion is the observations made in the practice. Especially the second derivative is calculated with up to 10% error on the edge nodes with polynomial fit approach.

The idea to correct this error is to add grid points where they are needed. One idea is to activate every grid point that corresponds to a basis location in 1-D in the direction of the stencil for every active grid point. Since the original 2-D basis locations are already active, the new grid points automatically have a stencil on the other direction than the direction of interest. The grid for this approach with $j_0 = 3$, $J = 7$, polynomial interpolation of order 4 for one active wavelet, $\psi_{7,7}^{6,3}$ can be seen in Figure 2.14. In Figure 2.14, first this approach is applied, then the minimum set is constructed. Even though this approach fixes some issues, it is seen in practice that it is not enough.

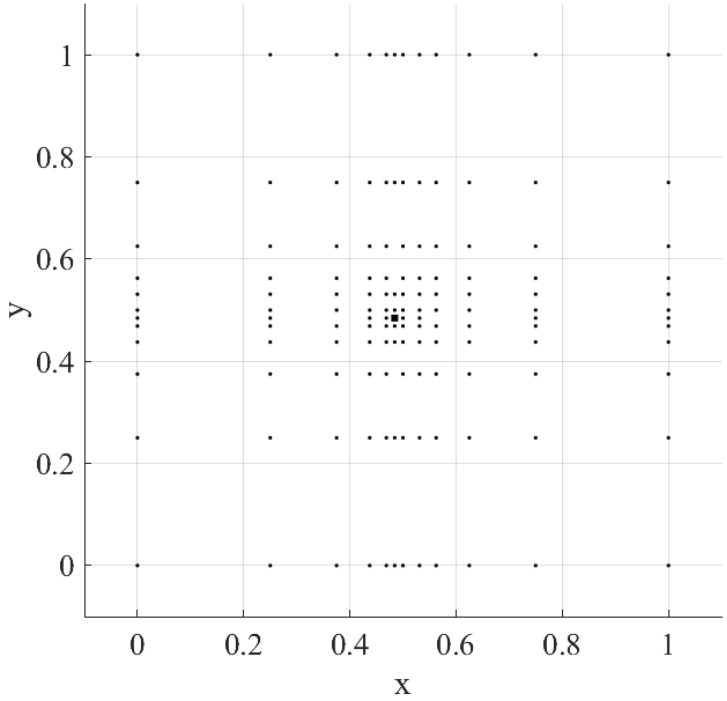


Figure 2.14. The first method for completing the stencils in the grid.

A more intense approach is to use type 1 or type 2 wavelets' minimum set criterion in Figure 2.10 as if they lie on a 1-D domain on each direction for every active grid point, and activate points that are spanned by the dyadic union of these perfect reconstructions. This way, we make sure that the grid points coming in with this operation have their stencil points on every direction, so they do not require any further attention. The application of this method is represented in Figure 2.15. For a given central point (x_i, y_j) and given points along y direction $\{x_{i-m}, x_{i-m+1}, ..., x_i, ..., x_{i+n-1}, x_{i+n}\}$ and along x direction $\{y_{j-k}, y_{j-k+1}, ..., y_j, ..., y_{j+l-1}, y_{j+l}\}$ a dyadic union of grid points is the set of grid points such that $\{(x_a, y_b) \mid a = i-m, ..., i+n \mid b = j-k, ..., j+l\}$.

A comparative figure for the minimum set construction, the first and second methods to complete the stencils in the grid can be seen in Figure 2.16. In the figure, the filled square is the only active wavelet on the grid, $\psi^{6,3}_{7,7}$, the dots are the minimum set, the $\times$s are the points added by the first method and the empty squares are the points added by the second method. In implementation, the second method and minimum set

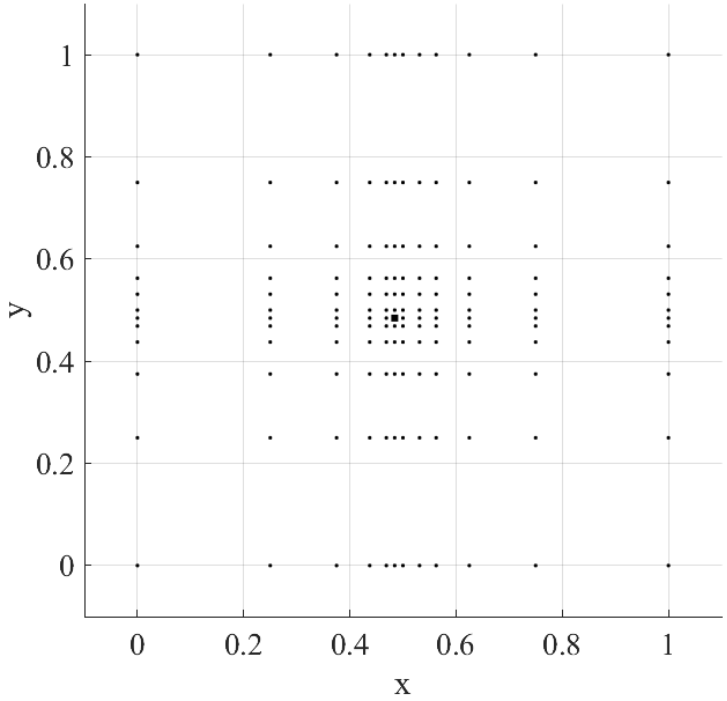


Figure 2.15. The second method for completing the stencils in the grid.

approach covers all the points of the other approaches anyway, so this path is chosen.

2.4.1. An Important Remark

The measures above must be taken for the stability of the finite difference differentiation. However, the stability issue is still not solved, but only reduced. Even after the application of the second method of grid completion, there are still point errors in the derivatives for polynomial fit approach. These errors show themselves especially at the regions where a transition from finer to coarser occur.

For wavelet parameters $j_0 = 4$, $J = 9$, 5^{th} order polynomial interpolation for wavelets (6 points for lifting scheme) and 4^{th} order polynomial interpolation for finite difference (5 point stencil on each direction), the Laplacian of function $f(x, y) = exp(-(\frac{x-0.25}{0.04})^2 - (\frac{y-0.5}{0.04})^2)$ on domain $[0, 1] \times [0, 1]$ is calculated on an irregular grid with $\epsilon = 10^{-5}$. The function and the irregular grid can be seen in Figure 2.17, and the error

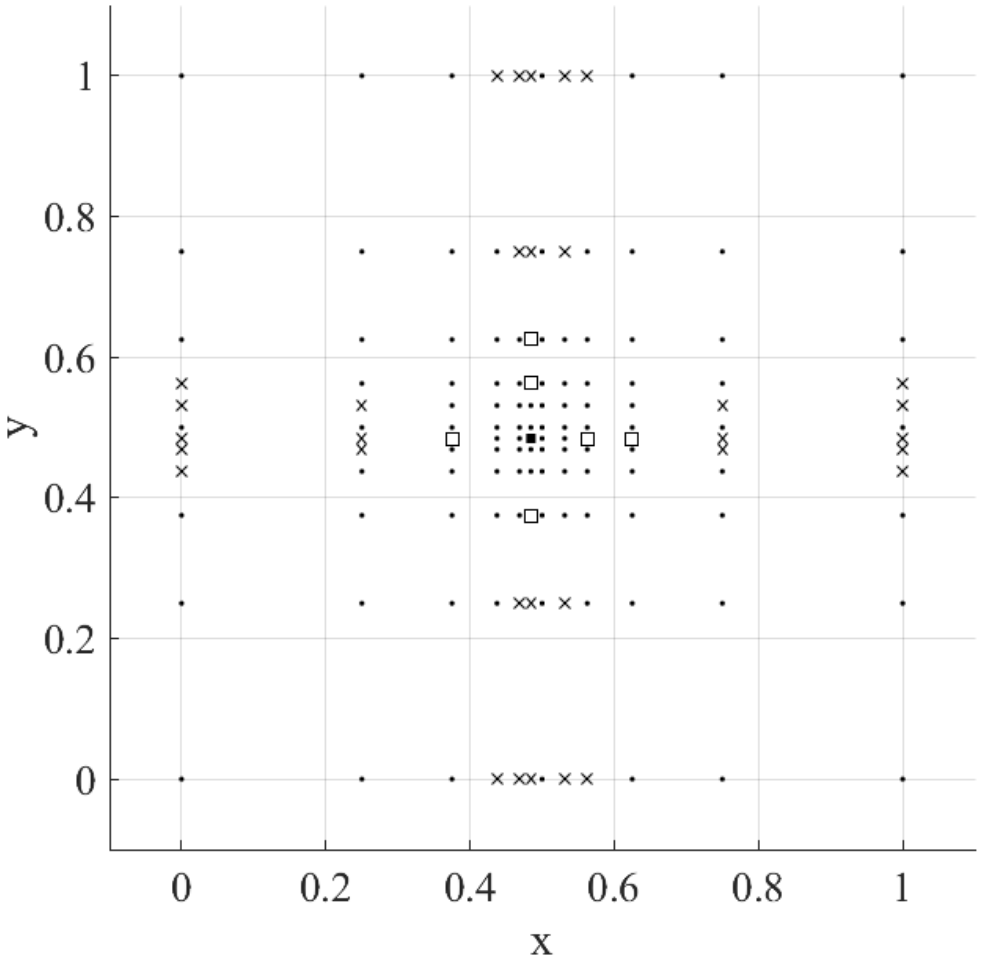
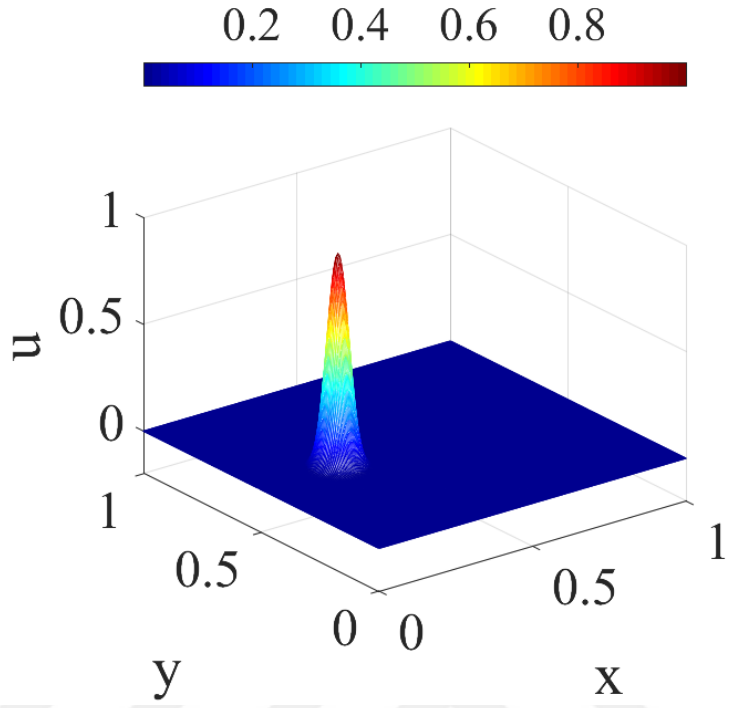
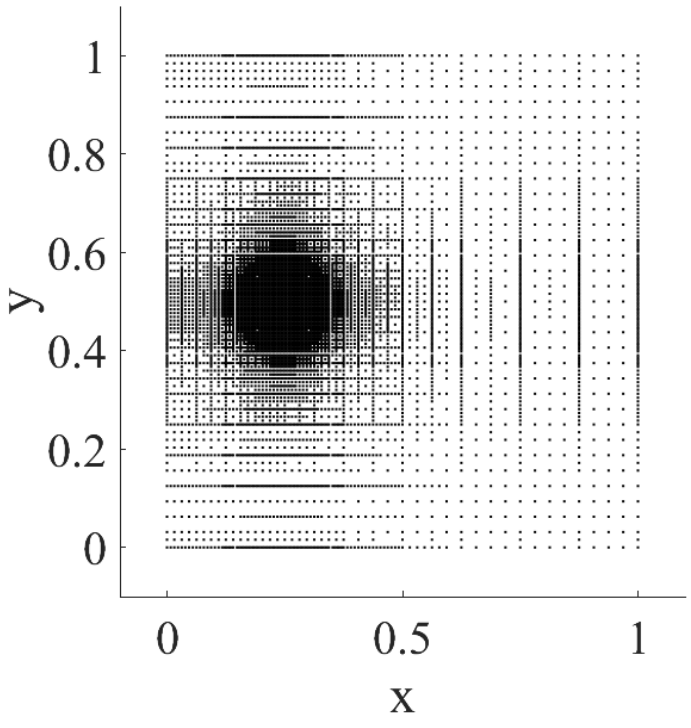


Figure 2.16. The comparison of the minimum set, first method and second method of completion of the grid. ■ = active wavelet, · = minimum set points, × = points added by first method, □ = points added by second method.



(a)



(b)

Figure 2.17. The sample function and the adapted grid.

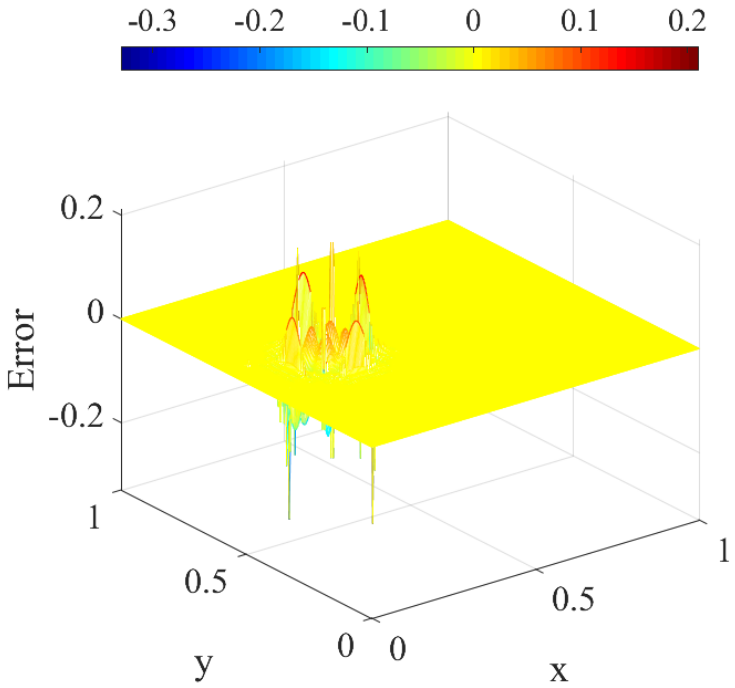


Figure 2.18. The error for the Laplacian of the sample function on the adapted grid.

can be seen in Figure 2.18.

The instabilities act as point sources on a Poisson Equation and point body forces in a Navier-Stokes Equation System. They cripple the accuracy of the solution, being multiplied by the time-step chosen for the time integration scheme. To remedy the effects of this instability, the time-step must be chosen with care. Smaller the time-step is, the less the effects of the instability are. Also, it is observed that once this instability affects a region, the wavelets sense this instability and make the grid finer around it for the next time-step of calculations. In a sense, wavelets fix the problem by themselves if the chosen time-step is small enough. This leads to oscillations of the active node number at a time-step. We do not develop any methods to determine the magnitude of the time-step in this work, but rather use trial and error to find a suitably small time-step.

3. NUMERICAL PROCEDURE AND SOLVED PROBLEMS

Having defined the wavelet transformation and the finite differences approximation on an irregular grid, we are ready to give the algorithm for the numerical procedure. It can be seen in algorithm 3.1

Having γ, wavelet transform get φ ;
Use φ to get irregular grid *grid* ;
Add neighbouring wavelets to *grid*;
Apply grid completion on *grid*
Apply Minimum Set Construction on *grid* ;
for $t = 0$ until t_{end} **do**
  Take a time step, g gets updated ;
  Wavelet transform on γ, get new φ ;
  Use new φ to get new *grid* ;
  Add neighbouring wavelets to new *grid*;
  Apply grid completion on *grid*
  Apply Minimum Set Construction on new *grid* ;
end for

Figure 3.1. Time integration scheme.

Here, γ represents the current interpolation of the solution function on the irregular grid, φ represents the matrix that holds the wavelet and basis function coefficients and *grid* represents the matrix that holds the irregular grid data. Taking the time step is done using the PETSc linear solvers. The function for the linear solver is constructed using the finite differences method explained above.

3.1. 1-D Poisson Equation

3.1.1. Moving Local Structure

The first problem is the 1-D transient Poisson equation. The equation is solved on a domain $[0, 1]$.

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} + f(x,t) \tag{3.1}$$

The exact solution is chosen to be equation 3.2

$$u(x,t) = exp\left(-\left(\frac{x-a-bt}{c}\right)^2\right) \tag{3.2}$$

The equation is solved with a consistent source function $f(x,t)$ and two Dirichlet boundary conditions. The solution is a Gaussian Pulse of height 1 moving on the computational domain from $x = 0.25$ to $x = 0.75$. The adaptive grid is expected to follow this motion.

Time integration is done using implicit scheme, i.e. Backwards-Euler, for both the second-order derivative and the source function. This can be seen in equation 3.3.

$$\frac{u_{n+1} - u_n}{\Delta t} = \frac{\partial^2 u_{n+1}}{\partial x^2} + f_{n+1} \tag{3.3}$$

3.1.2. Disappearing and Reappearing Local Structure

The second problem is also the 1-D transient Poisson equation. The equation is solved on a domain $[0, 1]$, time $t = [0, 2]$.

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} + f(x,t) \tag{3.4}$$

The exact solution is chosen to be equation 3.5

$$u(x,t) = (1-t)exp\left(-\left(\frac{x-a}{c}\right)^2\right) \tag{3.5}$$

The equation is solved with a consistent source function $f(x,t)$ and two Dirichlet boundary conditions. The solution is a Gaussian Pulse of height 1 first disappearing slowly on $x = a$, then reappearing until it reaches height -1 on the same spot . The adaptive grid is expected to slowly lose its fineness around $x = a$, the regain its fineness as the structure reappears. Time integration is defined by equation 3.3.

3.2. 2-D Poisson Equation

The first problem in 2-D is the transient Poisson equation with a non-zero source, on a square domain $[0, 1] \times [0, 1]$ represented by equation 3.6,

$$\frac{\partial u(x,y,t)}{\partial t} = \nabla^2 u(x,y,t) + f(x,y,t) \tag{3.6}$$

where the exact solution is chosen as the equation 3.7

$$u(x,y,t) = exp\left(-\left(\frac{x-a-bt}{c}\right)^2\right) exp\left(-\left(\frac{y-d}{c}\right)^2\right) \tag{3.7}$$

and the initial condition is 3.8.

$$u(x,y,0) = exp\left(-\left(\frac{x-a}{c}\right)^2\right) exp\left(-\left(\frac{y-d}{c}\right)^2\right) \tag{3.8}$$

The all Dirichlet boundary conditions and the forcing function $f(x,y,t)$ is chosen accordingly. Implicit method, i.e. Backwards Euler, is used for time integration and source function, as can be seen in equation 3.9.

$$\frac{u_{n+1}-u_n}{\Delta t} = \nabla^2 u_{n+1} + f_{n+1} \tag{3.9}$$

This problem is chosen because it is a good candidate to show the adaptive grid. The exact solution is a Gaussian Pulse of height 1 travelling on $y = 0.5$ line. The wavelet adaptive grid is expected to follow this bump throughout the solution.

3.3. 2-D Helmholtz Equation

Transient Helmholtz Equation with a non-zero source function is solved. The problem is solved on a square domain $[0,1] \times [0,1]$ from time $t = 0$ to time $t = 0.5$. The equation is represented as equation 3.10,

$$\frac{\partial u(x,y,t)}{\partial t} = \nabla^2 u(x,y,t) + u(x,y,t) + f(x,y,t) \tag{3.10}$$

where the chosen exact solution is chosen as 3.11.

$$u(x,y,t) = exp\left(-\left(\frac{x-a-bt}{c}\right)^2\right) exp\left(-\left(\frac{y-d}{c+\frac{t}{g}}\right)^2\right) \tag{3.11}$$

The initial condition, source function and Dirichlet boundary conditions on all four sides are chosen according to the exact solution 3.11. The exact solution is a

travelling and evolving local structure. It is a Gaussian Pulse that moves in positive x direction while its width increases along y axis. The wavelet adaptive grid is expected to adapt the solution with respect to the location and shape of the structure.

3.4. 2-D Isothermal Navier-Stokes Equation

Isothermal non-dimensional incompressible Navier Stokes Equations for a Newtonian fluid, equation 3.12, with continuity equation, equation 3.13, are solved on a square domain $[0,1] \times [0,1]$.

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\nabla p + \frac{1}{Re} \nabla^2 \vec{u} \tag{3.12}$$

$$\nabla \cdot \vec{u} = 0 \tag{3.13}$$

with Dirichlet boundary conditions,

$$\vec{u} = \alpha \tag{3.14}$$

where α is the possibly non-constant component of the Dirichlet boundary condition.

3.4.1. Time Integration Scheme

To integrate this system of equations in time, split-step method, also called projection correction method, is used as described by Kevlahan and Vasilyev [8]. According to Kevlahan and Vasilyev, the system of equations can be chopped into two parts, a projection step for velocity components which result in a non-solenoidal velocity field, i.e. a velocity field that does not obey the continuity equation 3.13, and a correction step which modifies this velocity into a field which obeys the continuity equation.

At a given time step, n having the current solenoidal (continuity obeying) velocity field $\vec{u_n}$, the projection step that gives the non-solenoidal velocity $\vec{u^*}$ is the following

equation 3.15.

$$\frac{\vec{u^*} - \vec{u_n}}{\Delta t} - \vec{u_n} \cdot \nabla \vec{u^*} = \frac{1}{Re} \nabla^2 \vec{u^*} \tag{3.15}$$

As it can be seen, the advection term is discretized semi-explicitly and the diffusion term is discretized implicitly. Thus, the resulting equations are linear. These are solved with PETSc's built-in Generalized Minimal Residual (GMRES) linear solver with Incomplete LU Factorization of 0 level of fill (ILU(0)) preconditioning.

The boundary conditions for $\vec{u^*}$ are identical with the boundary conditions of the original velocity $\vec{u}$. The pressure term is dropped altogether, although this is not ideal in terms of accuracy, the error that comes from this step is fixed by the correction step to some extent.

The correction step consists of two substeps. First substep is to solve a Poisson equation, equation 3.16.

$$\nabla^2 \Phi = \frac{1}{\Delta t} \nabla \cdot \vec{u^*} \tag{3.16}$$

with Neumann boundary conditions on all boundaries

$$\nabla \Phi \cdot \vec{n} = \vec{u^*} \cdot \vec{n} \tag{3.17}$$

where $\vec{n}$ is the normal vector pointing outwards from the boundary.

This equation with these boundary conditions is problematic most of the times, and requires special attention. This phenomenon will be explained in first subheading 3.4.1.1.

In the second step of the correction operation, having solved the Poisson Neumann Equation, Φ is modified and added to the non-solenoidal velocity field, equation 3.18, to have a $\vec{u_{n+1}}$ that satisfies the continuity equation.

$$\vec{u_{n+1}} = \vec{u^*} - \Delta t \nabla \Phi \tag{3.18}$$

The time integration step is completed at this point.

3.4.1.1. Poisson Neumann Equation. Suppose that there is a domain, Ω with boundaries $\partial\Omega$. A Poisson equation with a non-zero source function, equation 3.19 on this domain is defined

$$\nabla^2 \Phi + F = 0 \tag{3.19}$$

with Neumann boundary conditions on the boundary

$$\nabla \Phi \cdot \vec{n} = G \tag{3.20}$$

where $\vec{n}$ is the outward normal vector on $\partial\Omega$. For this equation to have a solution, the following condition has to be satisfied,

$$\int_{\Omega} F d\Omega = \oint_{\partial\Omega} G ds \tag{3.21}$$

where ds is the infinitesmal surface element on $\partial\Omega$. Otherwise, there is no solution to the equation.

This condition is called the compatibility condition. From a physical viewpoint, it can be explained as a heat source within the domain Ω. All the heat generated by the heat source F must be evacuated through the boundaries $\partial\Omega$ for the system to have a steady-state solution. Otherwise, the excess heat is constantly accumulated in

the domain Ω and there is no steady-state solution.

When 3.19 is discretized on a wavelet adapted grid, a system of linear equations is obtained, as can be seen in equation 3.22

$$A\vec{x} = \vec{b} \tag{3.22}$$

where A is the system matrix generated by the ∇^2 operator and the boundary conditions, $\vec{x}$ is the unknown vector that contains the discrete values of Φ and $\vec{b}$ is the right hand side vector generated by F and G.

The matrix A has one non-zero unit vector in its right and left null-spaces. In other words, there exists a non-zero right null-space vector $\vec{r}$ and a non-zero left null-space vector $\vec{l}$ such that

$$A\vec{r} = \vec{0} \tag{3.23}$$

$$\vec{l^T}A = \vec{0^T} \tag{3.24}$$

The discrete form of the compatibility condition is that the right hand side vector $\vec{b}$ and the left null-space vector $\vec{l}$ must be orthogonal, i.e. $\vec{b}$ must have no component in the subspace defined by $\vec{l}$.

$$\vec{b^T}\vec{l} = 0 \tag{3.25}$$

The right null-space vector is known to be $\vec{1}$, a vector of 1s. The left null-space vector is identical with the right null-space vector for symmetric matrices, but given the nature of the irregular grid this is almost never the case.

The compatibility condition is most generally not satisfied either, so the exact solution of equation 3.22 is not possible. To tackle this issue, there are different approaches. One is to calculate the left null space vector and remove its components fron right-hand side vector $\vec{b}$ as done by Wirasaet [19]. Another is to solve an augmented matrix using the already known right null-space vector $\vec{r}$ as performed by Henshaw [22]. One other idea is to have a Least-Squares Solution of the system, rather than the modification of the system.

The method used in this paper is to take advantage of the residual minimization property of GMRES. Since the residual of the next iteration of the GMRES must be less than or equal to the current iterate, it can be said that if the residual is not decreasing anymore, the GMRES has converged. The remaining part of the residual is on the left null-space of A, thus is not in the span of the Krylov Subspace. So, GMRES with no restart and ILU(5) preconditioning is used.

3.4.2. Lid Driven Cavity

Lid Driven cavity is a well-etablished benchmark problem for new methods. The problem is solved here using the split-step method explained above for the solution of equations 3.12 and 3.13. The solution domain is $[0, 1] \times [0, 1]$. The boundary conditions are as equation 3.26.

$$
\begin{aligned}
u(x,1) &= 1 \\
u(x,0) &= 0 \\
u(1,y) &= 0 \\
u(0,y) &= 0 \\
v(x,1) &= 0 \\
v(x,0) &= 0 \\
v(1,y) &= 0 \\
v(0,y) &= 0
\end{aligned}
\tag{3.26}
$$

Physically the problem is a cavity that has non-zero tangential velocity for one of its walls, in this case the top lid, where all other walls are stationary. No-slip boundary conditions are applied on all walls.

4. RESULTS AND DISCUSSIONS

4.1. 1-D Poisson Equation

4.1.1. Moving Local Structure

Using the coefficients of equation 3.2 as $a = 0.25$, $b = 1$, $c = 0.04$, time-stepping from $t = 0$ to $t = 0.5$ with timesteps $\Delta t = 2.5 \times 10^{-3}$, the results for two different spatial differentiations and regular finite differences are obtained.

Figure 4.1 represents the result obtained for polynomial fit differentiation. In the figure, the adaptive grid is decomposed to its levels to make it easier to interpret. The levels $j_0 + m$ represent all the levels of wavelets, only $j_0 - 1$ represents the basis function locations. As it can be clearly seen, the fineness of the grid follows the localized structure in the solution.

Figure 4.2 represents the comparison of the maximum errors during the time integration for polynomial fit differentiation, cubic spline differentiation and the regular finite differences analysis. It can be seen that the polynomial fit differentiation clearly overperforms the cubic spline differentiation. However, the performances of these two methods seem to be getting closer as node number increases, i.e. ϵ decreases. It can be seen that finite differences overperforms both approaches. The possible reason is that the Taylor Expansion based finite differences is naturally better at estimating derivatives. Also, this specific problem may not be a clear case, as the Gaussian Pulse may be too broad.

4.1.2. Disappearing and Reappearing Local Structure

Using the coefficients of equation 3.2 as $a = 0.25$, $c = 0.04$, time-stepping from $t = 0$ to $t = 2.0$ with timesteps $\Delta t = 5 \times 10^{-3}$, the results are obtained.

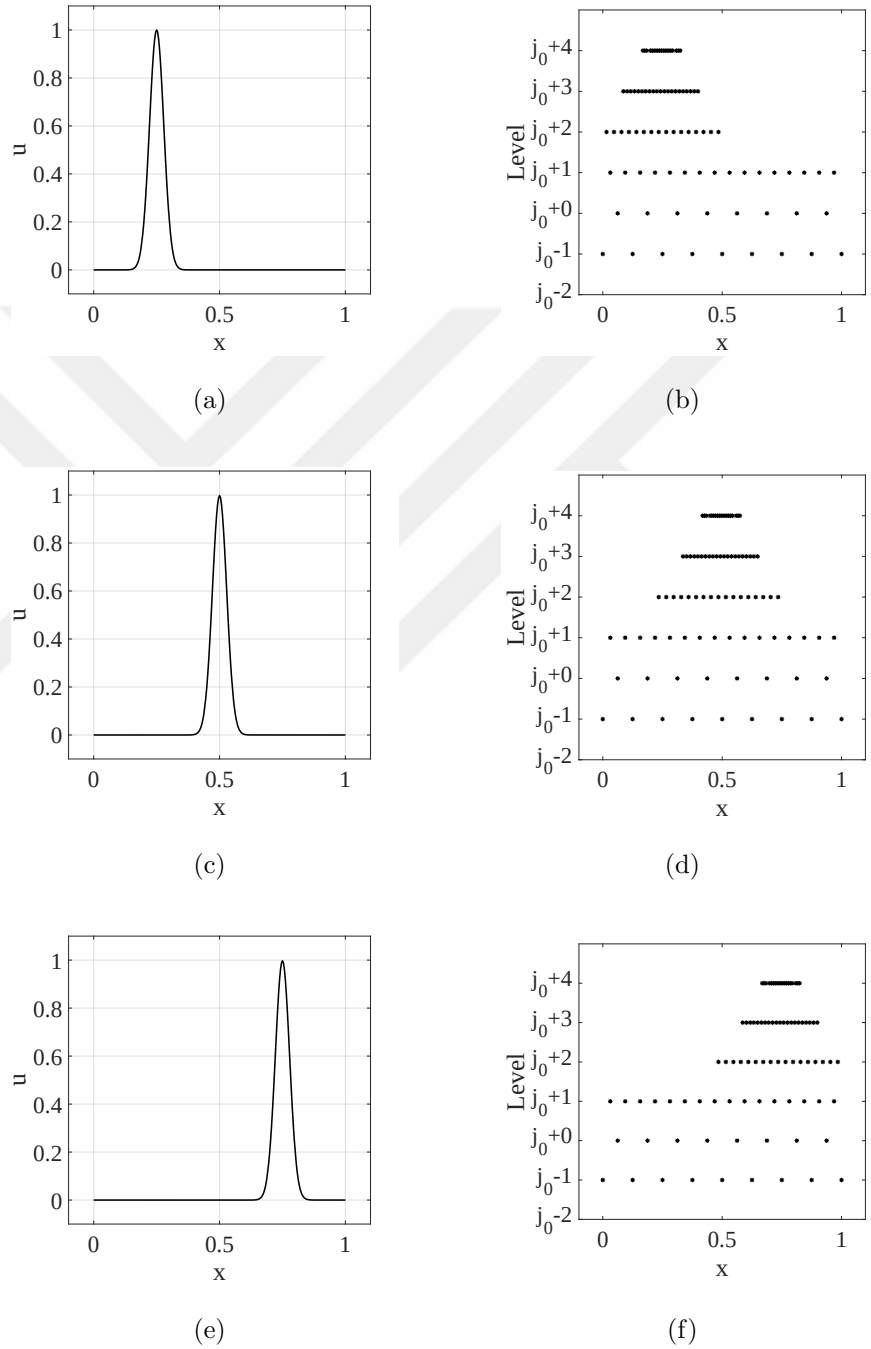


Figure 4.1. The time evolution of the solution (a,c,e) and the grid (b,d,f) for right travelling Gaussian Pulse at $t = 0$, $t = 0.25$, $t = 0.5$ from top to bottom.

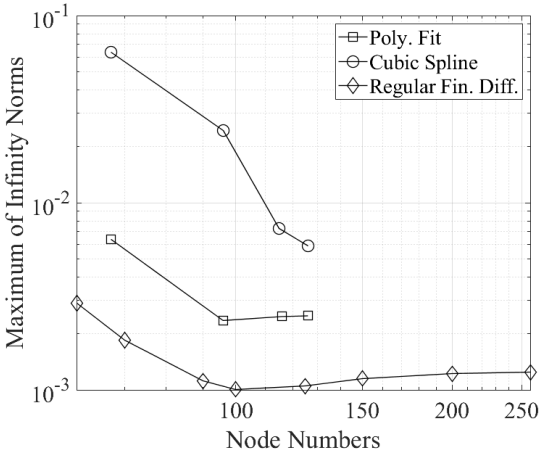


Figure 4.2. The error comparison between polynomial fit differentiation, cubic spline differentiation and regular finite differences for right travelling Gaussian Pulse.

In figure 4.3, it is clearly observed that the fineness of the adaptive grid around the localized structure first disappears as the structure itself disappears, then the fineness reappears as the structure reappears. This clearly demonstrates the ability of the wavelets to detect a new local structure.

The normalized Euclidian norm of vector of errors containing the maximum local error at each time-step vs. the node number is given at Figure 4.4. The normalized Euclidian norm is calculated with equation 4.1,

$$r = \sqrt{\frac{1}{N}\sum \varepsilon_i^2} \tag{4.1}$$

where ε_i is the maximum local error at i^{th} time-step and N is the number of timesteps taken at that run.

The Figure 4.4 shows that the errors are very close for regular finite differences and polynomial fit differentiation. This shows that the methods are very close, and in cases where the Gaussian pulse is narrower, WOFD is promising to outperform the regular finite differences. Cubic spline differentiation has higher error similar to the case for the first subheading 4.1.1. In these settings for 1-D, both for first subheading

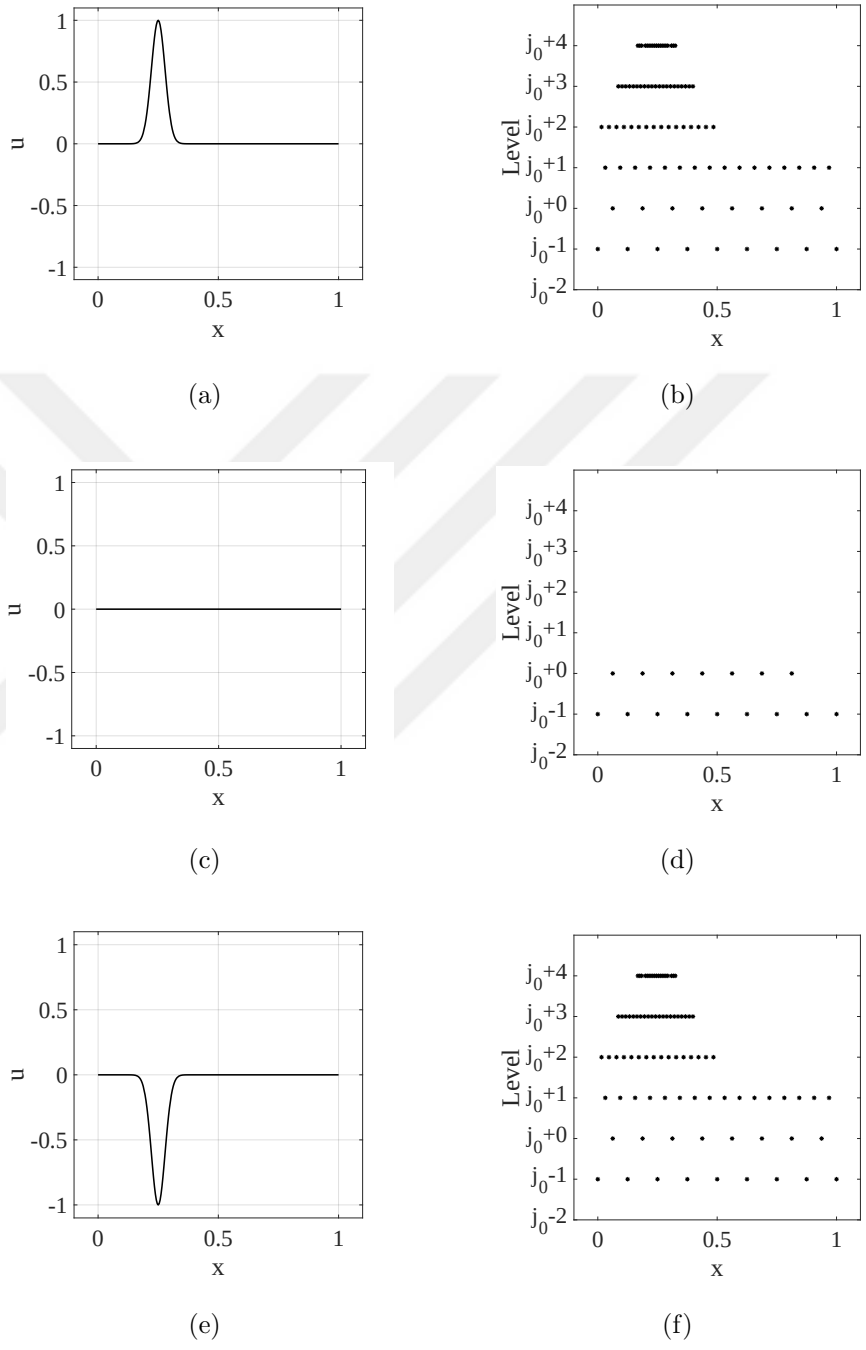


Figure 4.3. The time evolution of the solution (a,c,e) and the grid (b,d,f) for disappearing and reappearing Gaussian Pulse at $t = 0$, $t = 1.0$, $t = 2.0$ from top to bottom.

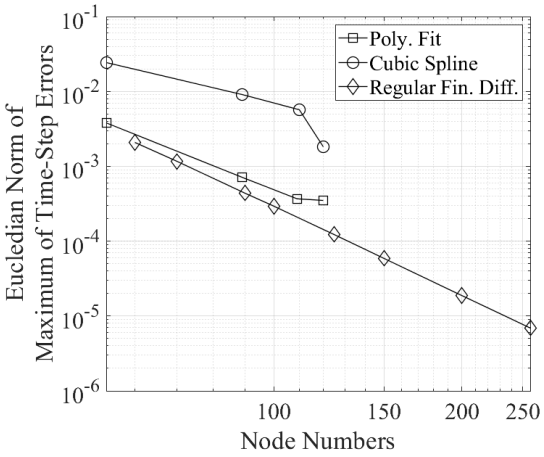


Figure 4.4. The error comparison between polynomial fit differentiation, cubic spline differentiation and regular finite differences for disappearing and reappearing Gaussian Pulse.

4.1.1 and this first subheading, it is more advantageous to use polynomial fits over cubic splines.

4.2. 2-D Poisson Equation

The exact solution, equation 3.7, is constructed with $a = 0.25$, $b = 1.0$, $c = 0.04$, $d = 0.5$. The Poisson Equation is solved from $t = 0$ to $t = 0.5$ with time steps $\Delta t = 2.5 \times 10^{-3}$. The wavelet parameters are as follows; $J = 9$, $j_0 = 4$, $\epsilon = 10^{-5}$, lifting scheme polynomials of order 5, finite difference polynomials of order 4. The solutions and the wavelet adaptive grids are seen in Figure 4.5.

In figure 4.5, it can be seen that the localized structure moves in positive x direction and the grid follows it as it is expected to. Note that there are no grid inner grid points without a neighbour to any of the four main directions. In other words, no inner grid point is an edge point in its own stencil.

More results are also taken with different values of ϵ, giving rise to different node numbers in the interpolation. Also, differentiations with cubic spline approximations

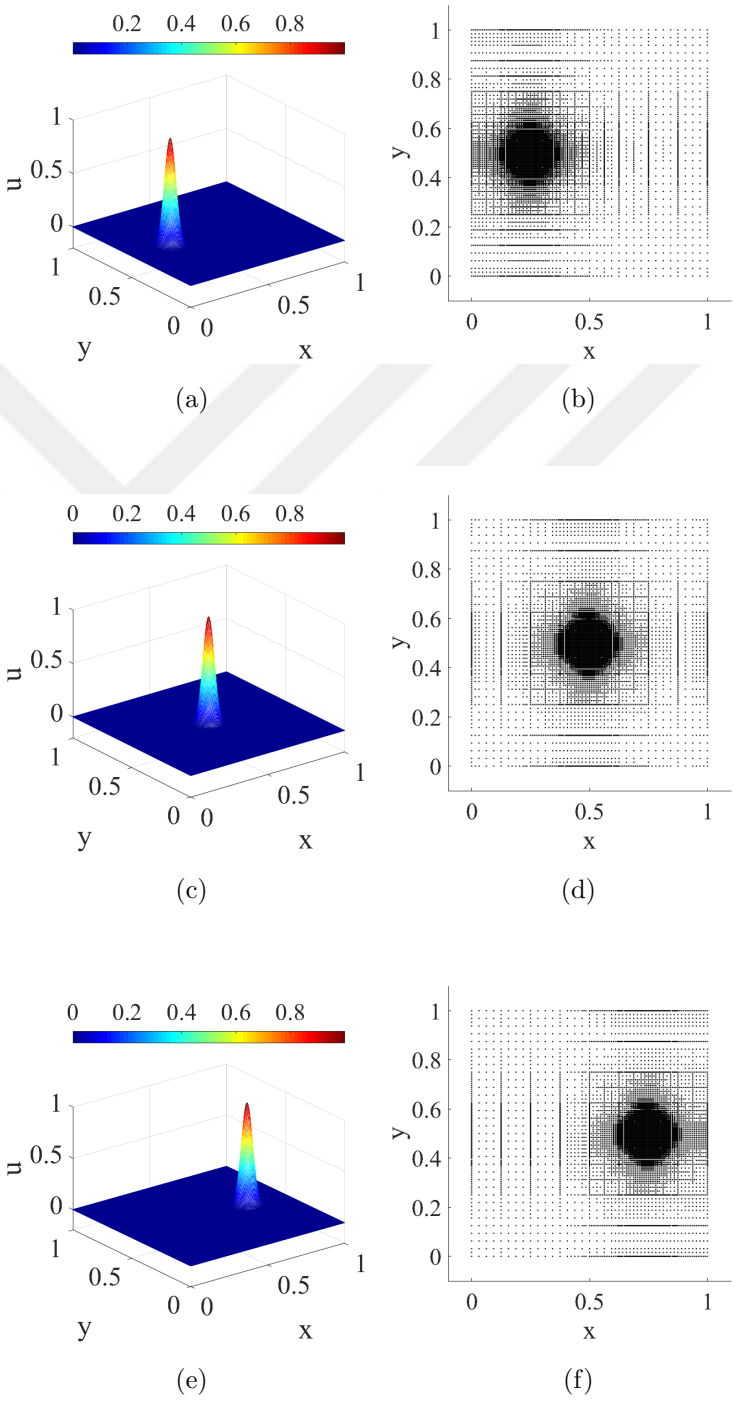


Figure 4.5. The solution(a,c,e) and the adaptive grid(b,d,f) for the Poisson Equation at $t = 0$, $t = 0.25$, $t = 0.5$ from top to bottom.

are performed. The maximums of the infinity norm errors of every time-step of a run are plotted in Figure 4.6. The node numbers correspond to ϵs of 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, and they are the maximum node numbers seen in their respective solutions. The maximum error is observed for $\epsilon = 10^{-4}$ in polynomial fit differentiation. This is most probably an outlier. Other than this outlier, the error seems to have a negative relationship with the node numbers, i.e. as the node number increases, the error decreases. This phenomenon can also be observed for the cubic spline based differentiation. The regular finite difference error seem to have a lowest value for node number of 15625. The rightmost value of it corresponds to the maximum fineness of the chosen wavelet parameters. It can be seen that the Wavelet Optimized Finite Differences (WOFD) with polynomial fit has almost the same error with regular finite differences when they have the same fineness level. However, WOFD with cubic splines clearly performs worse than both WOFD with polynomial fit and the regular finite differences in terms of fineness and node numbers.

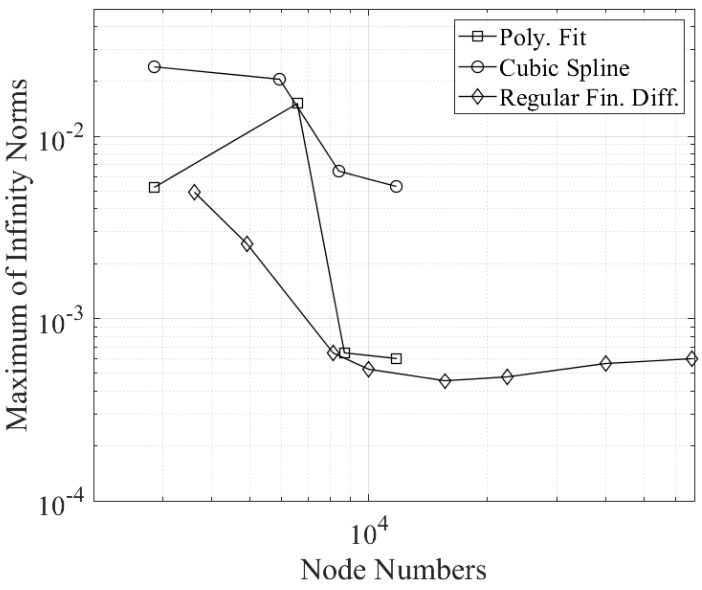


Figure 4.6. Error vs. node number for two kinds of the WOFD and regular finite differences. The line marked with $\circ$ is for cubic splines, the line marked with $\square$ is for polynomial fits and the line marked with $\lozenge$ is for regular grid finite differences.

In Figures 4.7 and 4.8, errors of the solution and active numbers are given with respect to time. Both data are given for $\epsilon = 10^{-4}$ and $\epsilon = 10^{-5}$ and for both polynomial fit and cubic spline differentiations. There are visible fluctuations in the node numbers timestep-by-timestep. This phenomenon is due to the point errors generated by the polynomial fit differentiation scheme. These point errors act as point source functions in the solution of the problems, thus create small bumps in their vicinities. These bumps are seen as irregularities by the wavelets and the grid is refined around them, until the bump is gone. Then the wavelets retreat, and the errors and the bump occur again. These fluctuations are also seen in error vs. time graph, Figure 4.7 for polynomial fit differentiation. The cubic spline differentiation seems to be not suffering from this phenomenon with regards to error, but only with regards to active node numbers. This may be because that the cubic spline errors are smoother, the irregularities created by them do not affect the infinity norm error. Thus, they do not appear on the error graph as much as polynomial fit differentiation errors.

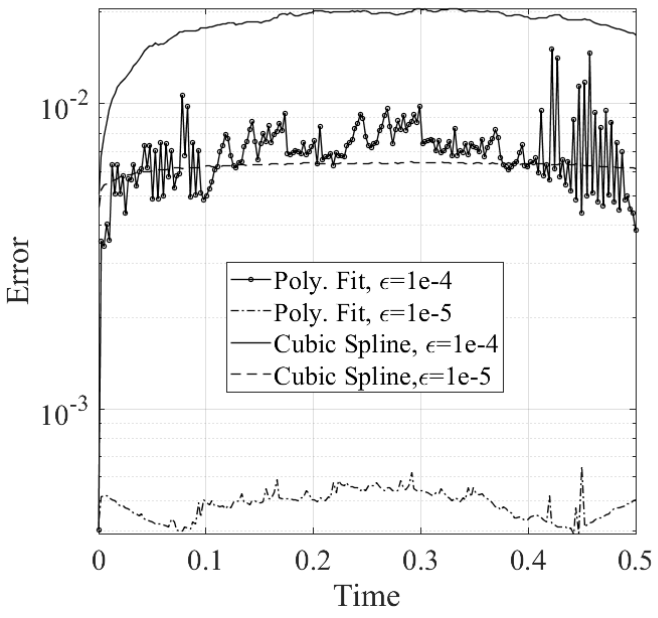


Figure 4.7. Error vs. time for two kinds of the WOFD for $\epsilon = 10^{-4}$ and $\epsilon = 10^{-5}$. – = cubic splines with $\epsilon = 10^{-4}$, - - = is cubic splines with $\epsilon = 10^{-5}$, the -○- = polynomial fit with $\epsilon = 10^{-4}$ and -·- = polynomial fit with $\epsilon = 10^{-5}$.

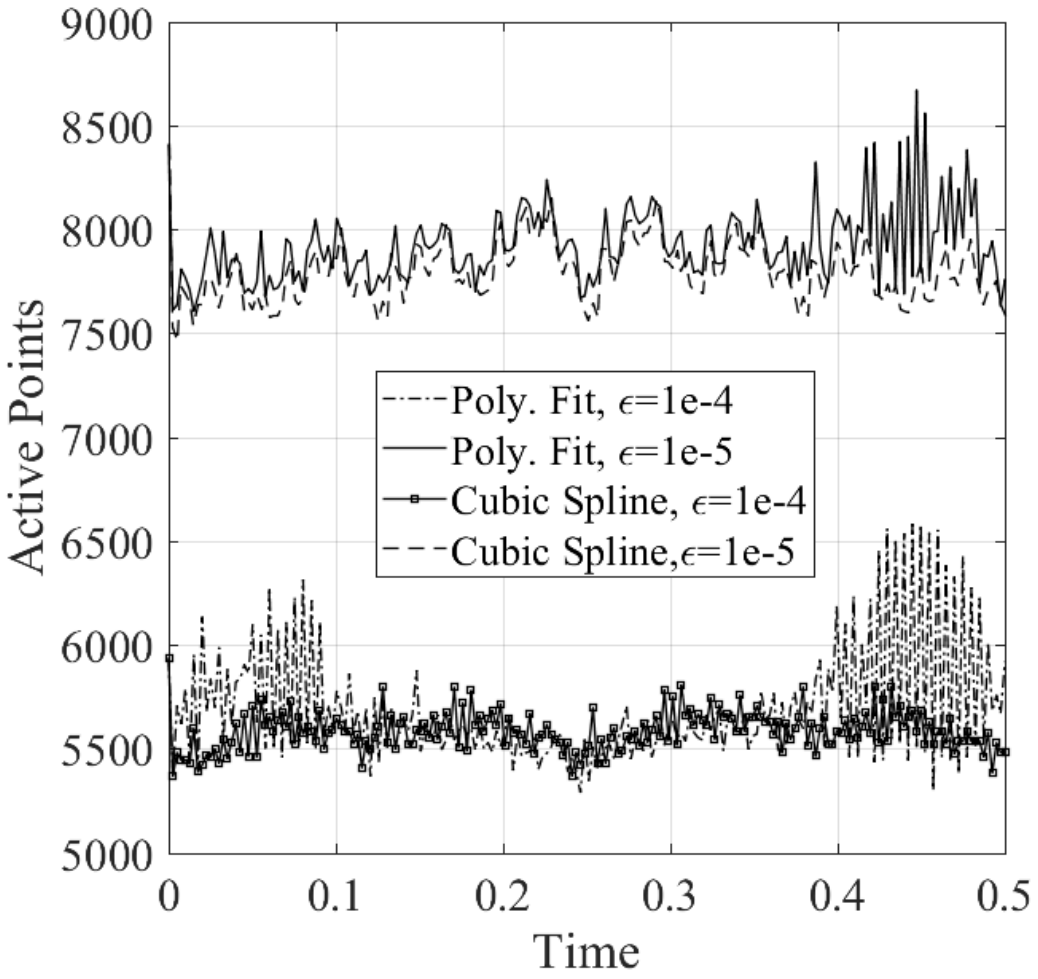


Figure 4.8. Number of active nodes vs. time for two kinds of the WOFD for $\epsilon = 10^{-4}$ and $\epsilon = 10^{-5}$. – = polynomial fit with $\epsilon = 10^{-5}$, - - = is cubic splines with $\epsilon = 10^{-5}$, -□- = cubic spline with $\epsilon = 10^{-4}$ and -·- = polynomial fit with $\epsilon = 10^{-4}$.

In Figure 4.9, the error vs. time graph is plotted for different sizes of 1-D stencils for $\epsilon = 10^{-5}$. The stencils numbers describe the stencils in 1-D. So when a 5 point stencil is mentioned, it is really a 9 point stencil in 2-D. The error for 7 point polynomial fit became too high, so the run was cut in the middle and the results are not represented here. The error for 3 points cubic spline is missing because the not-a-knot condition does not allow a cubic spline for 3 points. As it can be seen in the Figure 4.9, the error goes lower as the stencil size increases, until stencil size of 7 for polynomial fit. It can also be seen clearly that the fluctuations of the error is lower for 3 point stencil of polynomial fit differentiation compared to 5 point stencil. This suggests that the smooth error is dominant over the point errors of differentiation for 3 point stencils, so the instability caused by the point errors and the resulting ghost point sources is less effective.

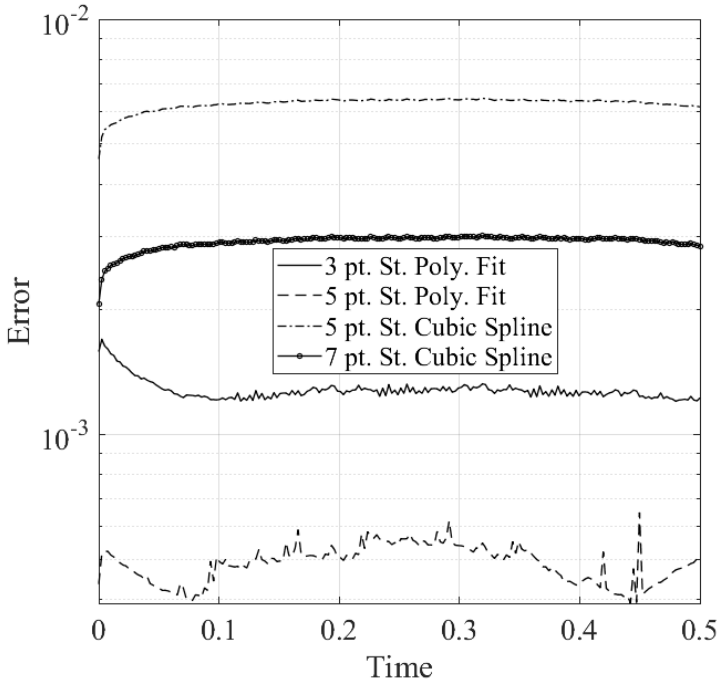


Figure 4.9. The error vs. time with different stencil sizes for Poisson Equation. – = polynomial fit with 3 point stencil, - - = polynomial fit with 5 point stencil, -·- = cubic spline with 5 point stencil, -○- = cubic spline with 7 point stencil.

4.3. 2-D Helmholtz Equation

The results for the moving and evolving structure produced by a Helmholtz Problem can be seen in Figure 4.10. The equation and wavelet analysis parameters are all the same as moving structure, with addition of $g = 10.0$. The time step is $\Delta t = 2.5 \times 10^{-3}$, and the time integration is carried from $t = 0$ to $t = 0.5$.

In Figure 4.10, it can be clearly seen that the wavelet adapted grid follows the motion and the shape of the localized structure. The compression of the grid risen almost steadily from 12% to 17.75% as the width of the structure increased. The fluctuations of the wavelet compression were also present, being around $\pm 0.5\%$ when the compression was 12% and around $\pm 1.25\%$ as the compression ratio goes to 17.75%.

The time-step independence of the solution is studied using time-steps of 10^{-3}, 2.5×10^{-3}, 5×10^{-3} and 10^{2} with $\epsilon = 10^{-5}$. The error vs. time plots can be seen in Figure 4.11. It can be seen that the errors reduce as the time-steps go to 10^{-3}. So, a compromise between error and total run-time is made and the following runs are taken with time-steps of 2.5×10^{-3}.

The comparison of errors of the solution by active node number for different spatial differentiation approaches and regular finite differences can be seen in Figure 4.12. For polynomial fit differentiation, it can be said that roughly the error decreases with decreasing ϵ. The peak error is recorded for one of the middle ϵs, 10^{-4}, however this is an outlier similar to the case with the Poisson equation solution. This is most probably due to the point errors of the polynomial fit differentiation scheme and its incompatibility with this specific problem and its parameters. The regular finite difference method seems to outperform both WOFD approaches. When the node number increases, thus the grid fineness increases, to a certain level for the regular finite differences, the time integration error becomes dominant and the error rate does not get lower. It is safe to say that the WOFD gives the same results with less nodes if the time integration error is the dominant error for the regular finite difference solution. Other than that, the superiority of regular finite differences is seen.

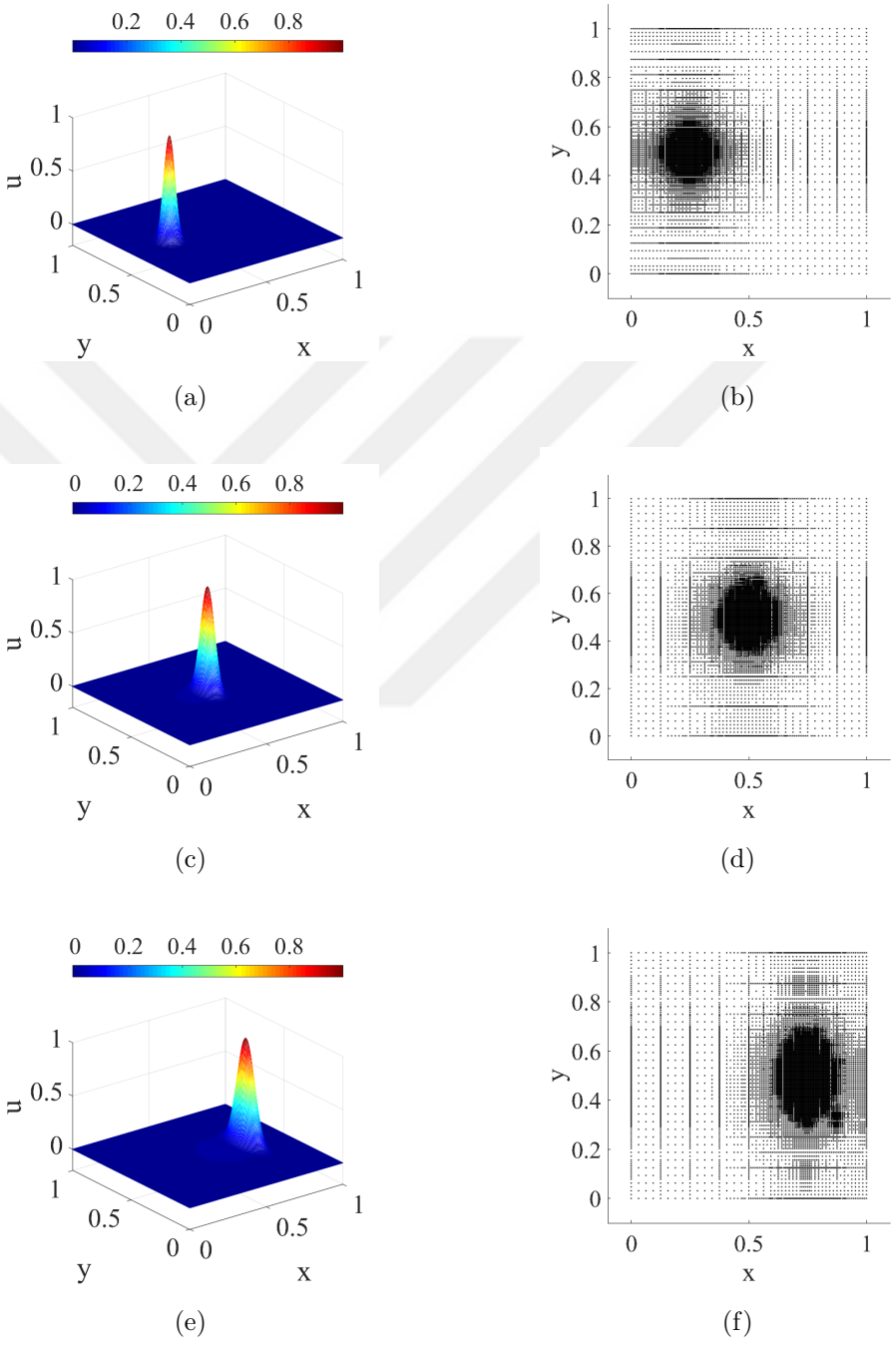


Figure 4.10. The solution(a,c,e) and the adaptive grid(b,d,f) for the Helmholtz Equation at $t = 0$, $t = 0.25$, $t = 0.5$ from top to bottom.

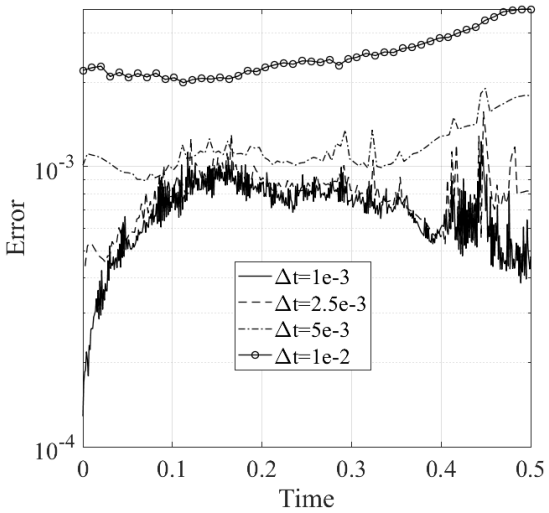


Figure 4.11. The error of the solutions vs. time with different Δt for Helmholtz Equation. – = time-step $\Delta t = 10^{-3}$, - - = time-step $\Delta t = 2.5 \times 10^{-3}$, -·- = time-step $\Delta t = 5 \times 10^{-3}$, -○- time-step $\Delta t = 10^{-2}$.

The reason cubic splines performs worse than polynomial fits is that the cubic splines have a higher error compared to polynomial fits. The error change vs. time data can be seen in Figure 4.13. In this figure, it can be seen clearly that the error is higher, but more stable for the cubic spline approach. The polynomial fit approach has great fluctuations in timestep-by-timestep errors, due to the point errors in the spatial differentiation process.

The errors vs. stencil sizes are also studied. The runs are taken with 1-D stencils of size 3,5 and 7 for polynomial fit approach. The run with stencil size of 7 was cut prematurely at 4^{th} time-step of 250. The reason was that the maximum value in the solution exceeded 1.5 (50% error).The runs were taken for 5 and 7 point 1-D stencils for cubic splines approach.The results comparing the error rates with stencil sizes can be seen in Figure 4.14. It can be seen that the errors are around half the magnitude with stencils size of 5 compared to 3 for polynomial fit approach. Also, given the large error of stencil size of 7, it can be concluded that the optimum size of the stencil is 5, finding the middle way between the accuracy and the locality of the derivative approximation.

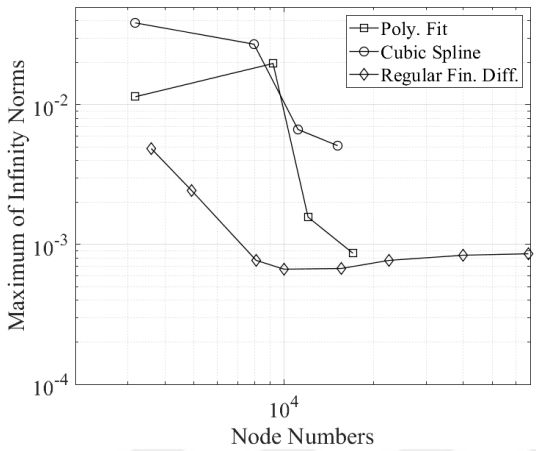


Figure 4.12. The error vs. node numbers. The line marked with ∘ is for cubic splines, the line marked with □ is for polynomial fits and the line marked with ◊ is for regular grid finite differences.

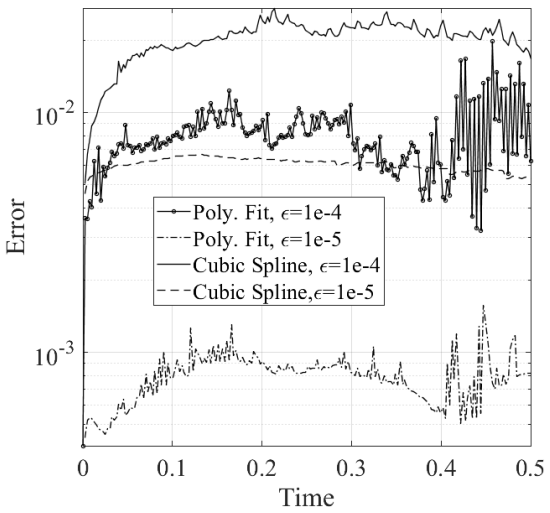


Figure 4.13. The error vs. time with different ϵ. – = cubic splines with $\epsilon = 10^{-4}$, - - = cubic splines with $\epsilon = 10^{-5}$, -∘- = polynomial fit with $\epsilon = 10^{-4}$, -·- = polynomial fit with $\epsilon = 10^{-5}$.

Regarding the cubic spline, stencil size of 7 seems to have a lower error than the 5 points stencil. The locality of the derivatives are not crippled by the cubic splines as much as they are crippled for the polynomial fits.

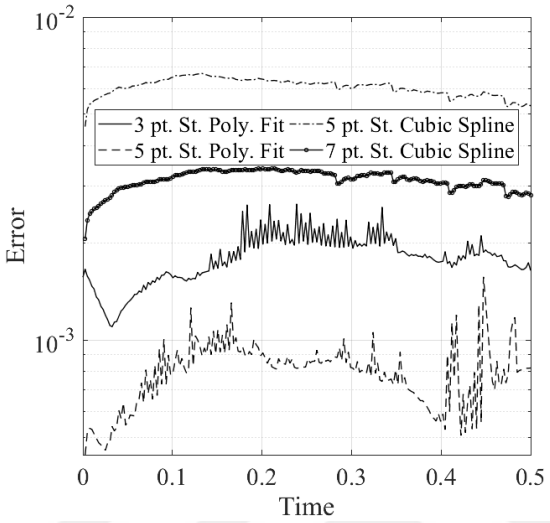


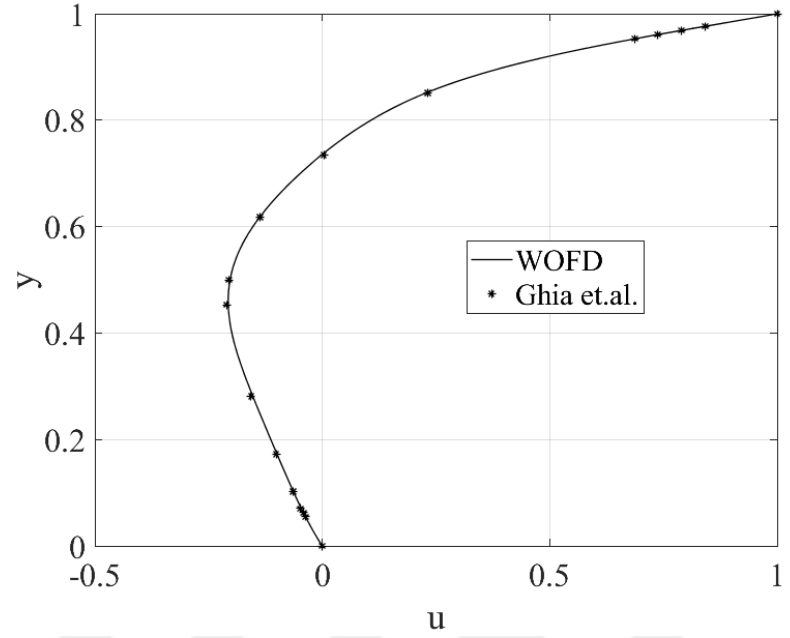
Figure 4.14. The error vs. time with different stencil sizes for Helmholtz Equation. – = polynomial fit with 3 point stencil, - - = polynomial fit with 5 point stencil, -○- = cubic spline with 5 point stencil, -·- = cubic spline with 7 point stencil.

4.4. 2-D Isothermal Navier-Stokes Equations

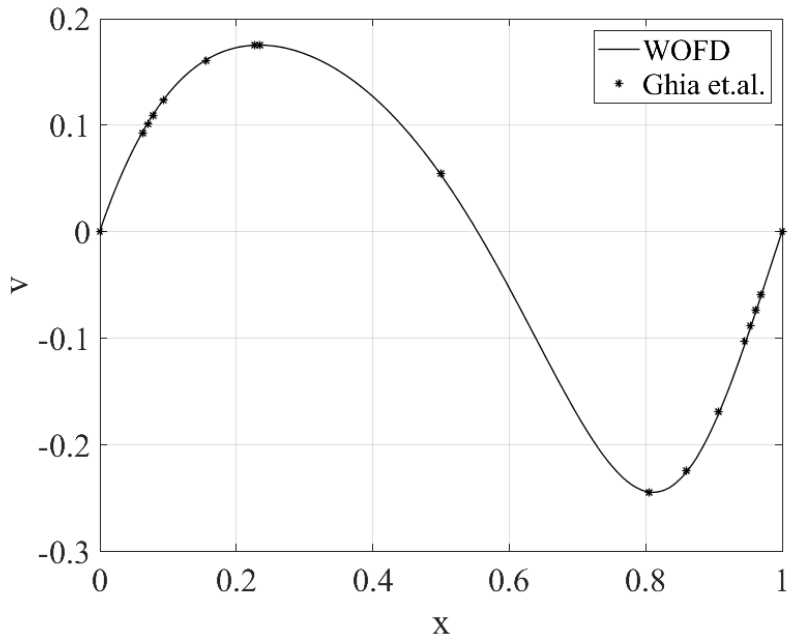
4.4.1. Lid Driven Cavity

The problem is solved with wavelet parameters $J = 9$, $j_0 = 4$, $Hr = 1$ and $Vr = 2$ and polynomial fit differentiation. Timesteps of $\Delta t = 10^{-2}$ is used. The streamlines and velocity profiles for $Re = 100$ and $Re = 400$ can be seen in Figures 4.15 through 4.17. The steady-states are taken at dimensionless time 20.0.

It can be seen that the quantitative and qualitative results for $Re = 100$ is in good-agreement with Ghia *et al.* [1]. On the other hand, for $Re = 400$ the quantitative results are not in good-agreement. The most probable reason is that the time-step chosen is too high to resolve this Re accurately.

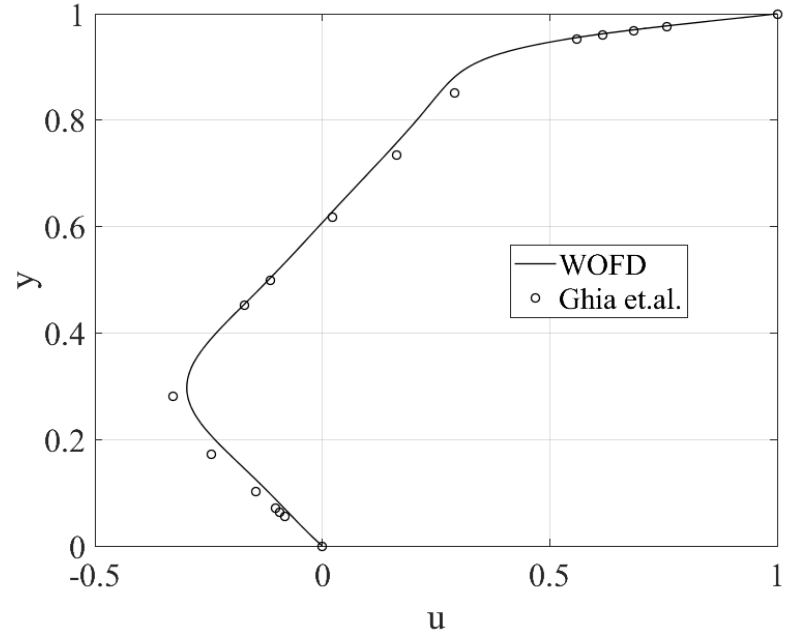


(a)

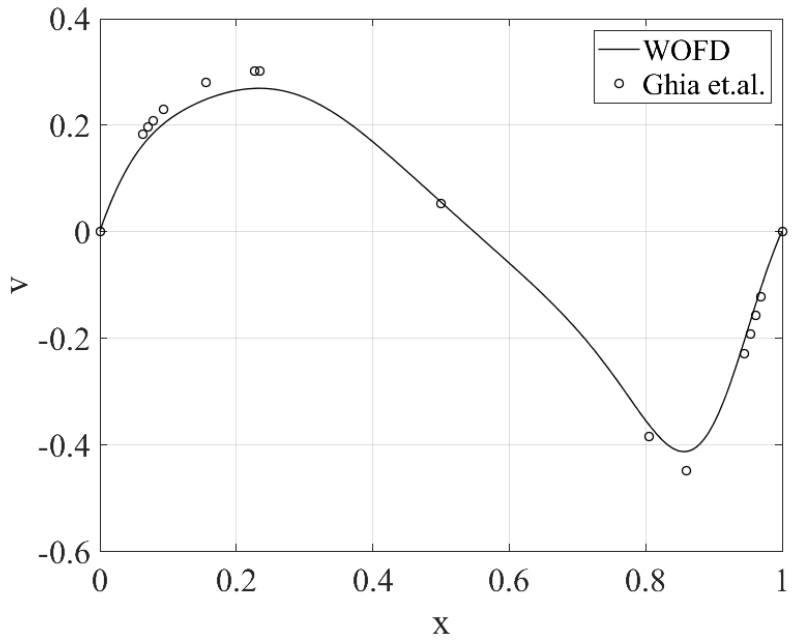


(b)

Figure 4.15. The u vs. y profile on $x = 0.5$ (a) and the v vs. x profile on $y = 0.5$ (b) for steady-state $Re = 100$ and comparison to [1].

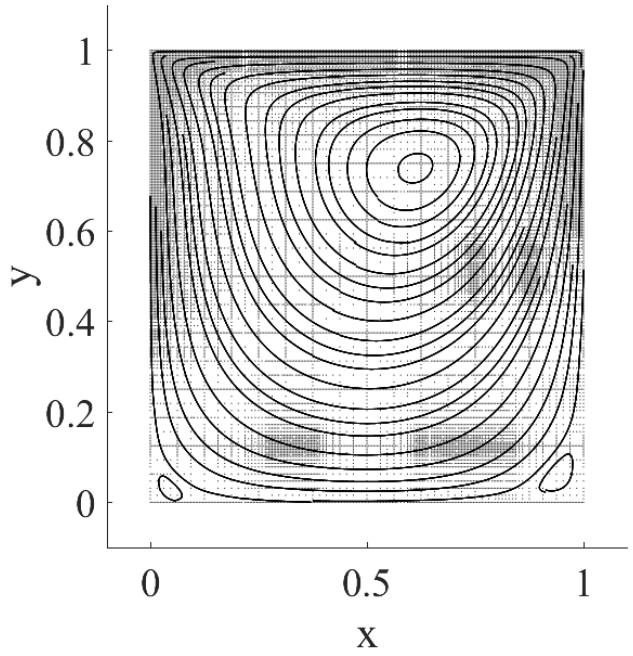


(a)

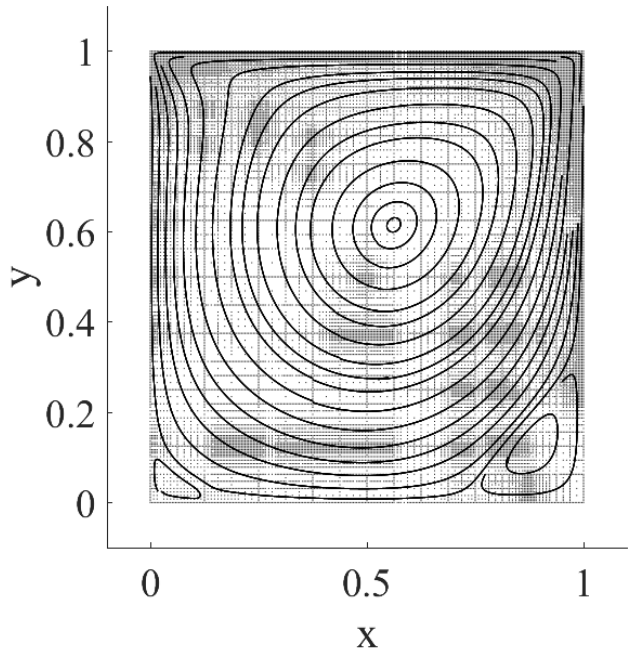


(b)

Figure 4.16. The u vs. y profile on $x = 0.5$ (a) and the v vs. x profile on $y = 0.5$ (b) for steady-state $Re = 400$ and comparison to [1].



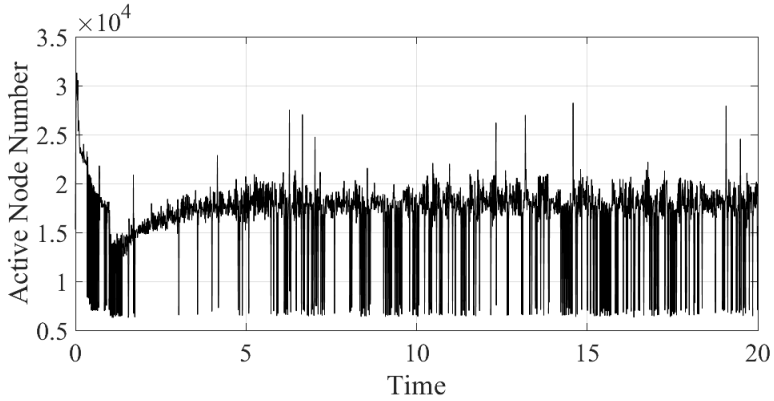
(a)



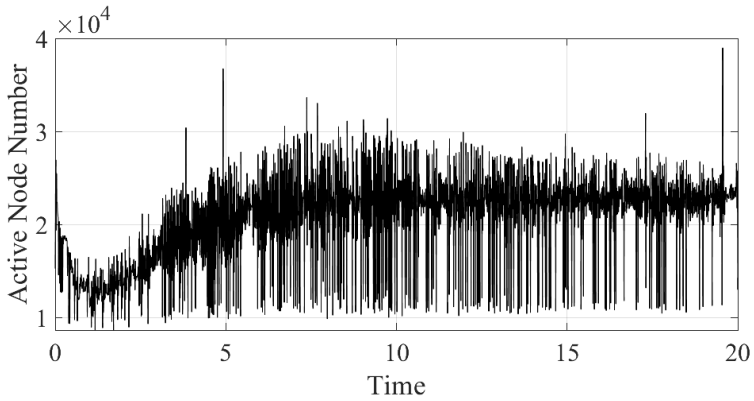
(b)

Figure 4.17. The streamline patterns and the adaptive grid for steady state for $Re = 100$ (a) and $Re = 400$ (b).

The adaptive grids can be seen in Figure 4.17. It is noticable that the grid density increases along the edges of the moving boundary. This is expected due to the singularity and sharp velocity direction changes in the corners. It can also be seen that there are denser grid areas well within the flow field, where no sharp changes occur. This is due to the point error of the polynomial fit differentiation, and wavelet's efforts to fix it. In this case, the point errors act as artificial body forces, creating small disturbances in the flow field and these disturbances are eliminated by wavelets in the next time steps. This is seen more intensely in $Re = 400$ case, which suggests that the increased non-linearity in the original problem affects the differentiation accuracy negatively. One possible way to eliminate this is to use smaller time-steps.



(a)



(b)

Figure 4.18. The active node number vs. time for $Re = 100$ (a) and $Re = 400$ (b).

The fluctuations in the number of active points are much higher than Poisson and Helmholtz Equations. The plots for active node numbers vs. time can be seen in Figure 4.18. It can be seen that the average node number is around 18000 for $Re = 100$ and around 22000 for $Re = 400$. The amplitude of fluctuations is about 12000 for $Re = 100$ and 20000 for $Re = 400$. This suggests that the wavelet compression for $Re = 100$ is better, but the ratio of fluctuations and average node numbers are also higher for $Re = 100$. Lower Re seems to be more grid-wise unstable about point errors arising from the differentiation procedure compared to the higher Re.

5. CONCLUSIONS

Wavelet analysis is a multi-resolution tool for function interpolations. The main notion is to represent the given function as the superposition of compact functions of different levels of resolution, namely wavelets. It is observed that wavelets on the finer scales have higher multiplying coefficients when there is an irregularity, such as a jump, in the function. Otherwise, these finer scale wavelets have insignificant coefficients. By removing these wavelets with insignificant coefficients and their associated grid points, an adaptive grid is obtained.

To utilize the adaptive grid for spatial derivative calculations, there are three main methods: Wavelet Galerkin, Wavelet Collocation, and Wavelet Optimization. In this work, Wavelet Optimization is used with the specific branch Wavelet Optimized Finite Differences (WOFD). On the irregular grid, two variants of finite difference discretization are considered. One variant is based on local polynomial fits and one variant is based on cubic splines. The main idea of both variants is to apply a polynomial fit or a cubic spline on the irregular stencil. Then this spline or fit is differentiated analytically to get the derivative values. These approaches are compared and it was seen that the error is higher and smoother for the cubic splines and lower but rougher for the polynomial fits. It is seen that the rough errors of the polynomial fits are fixed by wavelets on each step of the time-integration, so they are tolerable if a sufficiently small time-step is used.

The proposed adaptive grid with finite differences is applied to several engineering problems. First, 1-D problems are analyzed. Two 1-D Poisson Equations with known exact solutions, one a moving Gaussian Pulse and other a disappearing and reappearing Gaussian Pulse, are solved. It is seen in the former case that the wavelets refine the grid following the location of the pulse. In the latter case, it is seen that the grid fineness disappeared as the pulse in the solution died off, then the grid fineness reappeared with the pulse coming back. In both cases, the natural ability of wavelets to generate an adaptive grid is observed clearly. The WOFD could not perform better than the

regular finite differences in either case. It is concluded that this is due to the nature of the exact solutions, if a rougher exact solution were chosen the WOFD would have a chance of creating more accurate solutions than regular finite differences.

Several problems are examined in 2-D. 2-D Poisson Equation and 2-D Helmholtz Equations are solved for different known exact solutions. The Poisson equation has a 2-D moving Gaussian Pulse as an exact solution. The wavelet grid adaptation is observed. A good accuracy is accomplished using only around 12% of the maximum number of possible grid points. However, this accuracy is also observed with regular finite differences with same number of grid points, most probably due to the nature of the solution. The Helmholtz problem has a moving and evolving 2-D Gaussian Pulse as the exact solution. The wavelet grid adaptation is observed and the compression of the grid is between 12% and 17.75%. Again, the solution is not rough enough to show the advantages of WOFD.

In both 2-D problems, step-by-step oscillations are seen in the number of the grid points, due to the attempt of wavelets to fix the point errors of the finite-difference discretization. It is observed that the grid fineness increases to fix the issues in the region where these errors exist. Once the issue is fixed, the grid goes back to coarser state, giving rise to the errors again. This phenomenon is observed to cause cycles.

Finally, 2-D lid-driven cavity problem is solved. A split-step (projection-correction) scheme is adopted for the solution of the Navier-Stokes equations in primitive variables. This is performed to show an alternative use for WOFD. The qualitative results are in agreement with the literature, but the quantitative results agrees with the literature only for lower Re. The reason for this was the magnitude of the chosen time-step. The time-step is larger than needed for high Reynolds Number, Re. The step-by-step fluctuations in the number of grid points are also observed, even in greater magnitudes than Poisson and Helmholtz Problems. It is commented that the point errors act as artificial body-forces, disturbing the flow.

Overall, it is concluded that the wavelet compression is an easy-to-use tool for grid adaptation. However, finite differences proves to be difficult on an irregular grid. The discretizations causes undesired point errors. Trying to fix those errors, the wavelet grid-adaptation results in step-by-step fluctuations of the irregular grid, crippling the effectiveness of the wavelet compression.

The results obtained here in this study are not necessarily indicative of the superiority of the method. However, our intention in developing the method has been to develop a tool for applications in turbulence, in particular. Future studies may demonstrate the effectiveness of the method. The future work suggestions are to apply WOFD to higher Reynolds Numbers with smaller time-steps and possibly apply trigonometric and exponential polynomials for spatial discretization. Also, Poisson-Neumann Equation of the projection-correction scheme can be investigated more deeply.

REFERENCES

1. Ghia, U., K. N. Ghia and C. T. Shin, "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method", *Journal of Computational Physics*, Vol. 48, No. 3, pp. 387–411, 1982.

2. Vasilyev, O., S. Paolucci and M. Sen, "A multilevel wavelet collocation method for solving partial differential equations in a finite domain", *Journal of Computational Physics*, Vol. 120, No. 1, pp. 33–47, 1995.

3. Vasilyev, O. V. and S. Paolucci, "A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain", *Journal of Computational Physics*, Vol. 125, No. 2, pp. 498–512, 1996.

4. Vasilyev, O. V., D. A. Yuen and S. Paolucci, "Solving PDEs Using Wavelets", *Computers in Phys.*, Vol. 11, No. 5, pp. 429–435, 1997.

5. Vasilyev, O. V. and C. Bowman, "Second-Generation Wavelet Collocation Method for the Solution of Partial Differential Equations", *Journal of Computational Physics*, Vol. 165, No. 2, pp. 660–693, 2000.

6. Vasilyev, O. V., "Solving multi-dimensional evolution problems with localized structures using second generation wavelets", *International Journal of Computational Fluid Dynamics*, Vol. 17, No. 2, pp. 151–168, 2003.

7. Vasilyev, O. V. and N. K. Kevlahan, "Hybrid wavelet collocation – Brinkman penalization method for complex geometry flows", *International Journal for Numerical Methods in Fluids*, Vol. 538, No. May 2001, pp. 531–538, 2002.

8. Kevlahan, N. K. R. and O. V. Vasilyev, "An adaptive wavelet collocation method for fluid-structure interaction at high Reynolds numbers", *Siam Journal on Scientific Computing*, Vol. 26, No. 6, pp. 1894–1915, 2005.

9. Goldstein, D. E. and O. V. Vasilyev, "Stochastic coherent adaptive large eddy simulation method", *Physics of Fluids*, Vol. 16, No. 7, pp. 2497–2513, 2004.

10. Regele, J. D. and O. V. Vasilyev, "An adaptive wavelet-collocation method for shock computations", *International Journal of Computational Fluid Dynamics*, Vol. 23, No. 7, pp. 503–518, 2009.

11. Jameson, L., "On the wavelet optimized finite difference method", *ICASE Report 94-9*, NASA CR-191601, 1994, https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19950015740.pdf.

12. Jameson, L., "A Wavelet-Optimized, Very High Order Adaptive Grid and Order Numerical Method", *SIAM Journal on Scientific Computing*, Vol. 19, No. 6, pp. 1980–2013, 1998.

13. Walden, J., "Filter Bank Methods for Hyperbolic PDEs", *SIAM Journal on Numerical Analysis*, Vol. 36, No. 4, pp. 1183–1233, 1999.

14. Holmström, M., "Solving Hyperbolic PDEs Using Interpolating Wavelets", *SIAM Journal on Scientific Computing*, Vol. 21, No. 2, pp. 405–420, 1999.

15. Jameson, L. and T. Miyama, "Wavelet Analysis and Ocean Modeling: A Dynamically Adaptive Numerical Method 'WOFD-AHO' ", *Monthly Weather Review*, Vol. 128, No. 5, pp. 1536–1549, 2000.

16. Cruz, P., A. Mendes and F. D. Magalhães, "Wavelet-based adaptive grid method for the resolution of nonlinear PDEs", *AIChE Journal*, Vol. 48, No. 4, pp. 774–785, 2002.

17. Santos, J. C., P. Cruz, F. D. Magalhães and A. Mendes, "2-D wavelet-based adaptive-grid method for the resolution of PDEs", *AIChE Journal*, Vol. 49, No. 3, pp. 706–717, 2003.

18. Yousefi, H., A. Noorzad A. and J. Farjoodi, "Simulating 2D Waves Propagation in Elastic Solid Media Using Wavelet Based Adaptive Method", *Journal of Scientific Computing*, Vol. 42, No. 3, pp. 404–425, 2009.

19. Wirasaet, D., *Numerical Solutions Of Multi-Dimensional Partial Differential Equations Using An Adaptive Wavelet Method*, Phd. dissertation, University of Notre Dame, 2007.

20. Balay, S., S. Abhyankar, M. F. Adams, *et al.*, *PETSc Web Page*, 2018, `http://www.mcs.anl.gov/petsc`, accessed at June 2018.

21. Balay, S., S. Abhyankar, M. F. Adams, *et al.*, *PETSc Users Manual*, Tech. Rep. ANL-95/11 - Revision 3.9, Argonne National Laboratory, 2018, `http://www.mcs.anl.gov/petsc`.

22. Henshaw, W., "A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids", *Journal of computational physics*, Vol. 113, No. 1, pp. 13–25, 1994.