The Pennsylvania State University
The Graduate School

GRAPH-BASED TOPICS IN APPLIED MATHEMATICS

A Thesis in
Mathematics
by
John C. Urschel

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2013

The thesis of John C. Urschel was reviewed and approved* by the following:

Victor Nistor
Professor of Mathematics
Thesis Co-Advisor

Ludmil Zikatanov
Professor of Mathematics
Thesis Co-Advisor

Svetlana Katok
Professor of Mathematics

*Signatures are on file in the Graduate School.

Abstract

Computational problems in applied mathematics often have strong ties to graph/grid based structures. We treat graph-based problems in applied mathematics. We consider the spectral bisection problem for general graphs, and extend Miroslav Fiedler's results to more general graphs, using the theory of irreducible matrices. In addition, we apply the spectral bisection theory to graph partitioning, and create a cascadic Multigrid algorithm to solve for the Fiedler vector of a graph Laplacian, thus producing a bisection. We give convergence results for a sub class of graphs for our method. Finally, we consider the numerical computation of the fair price of barrier options. In practice, this becomes a problem of solving a partial differential equation numerically. We introduce a technique where the grid is treated in space and time simultaneously. We use a Multigrid V-cycle, in which the coarse grids are chosen adaptively, based on the properties of the finer structured graph.

Table of Contents

List of Figures

List of Tables

Acknowledgments

The author recieved a significant amount of support from a number of sources. The author is grateful to have had such great supervisors such as Victor Nistor, Jinchao Xu, and Ludmil Zikatanov. In addition, without the help and guidance of Xiaozhe Hu, this thesis would not have been possible.

Dedication

To Venita and John

Chapter 1

Introductory Comments

This thesis treats graph-based problems in applied mathematics. We treat three very different graph-type problems. We begin by treating the spectral bisection problem for general graphs, and extend Miroslav Fiedler's results to more general graphs in Chapter 2. In Chapter 3, we apply the spectral bisection theory to graph partitioning, and create a multilevel algorithm to solve for the Fiedler vector of a graph Laplacian, thus producing a bisection. Finally, in Chapter 4, we consider the numerical computation of the fair price of barrier options. In practice, this becomes a problem of solving a partial differential equation numerically. We introduce a technique where the grid is treated in space and time simultaneously. We use a multilevel structure, in which the coarse grids are chosen adaptively, based on the properties of the finer structured graph. We give a more detailed overview of each chapter below.

1.1 Theory of Connected Spectral Partitioning

The theorems of Miroslav Fiedler regarding bisections of irreducible matrices lies at the heart of graph partitioning theory. We consider the framework of graph bisections, in reference to the Fiedler vector. We extend the class of graphs that are guaranteed to produce connected spectral bisections. In addition, we show that for all connected graphs there exists a Fiedler vector such that connectedness is preserved by the bisection. We find that the subset of such Fiedler vectors has positive measure.

1.2 Cascadic Multigrid as an Eigensolver for Fiedler Vectors

The Fiedler vector of a graph Laplacian is the eigenvector corresponding to the second smallest eigenvalue. This vector has been found to have applications in fields such as graph partitioning and graph drawing. We develop a cascadic multigrid algorithm for fast computation of this vector. It is a purely algebraic approach based on a heavy edge coarsening scheme and a modified form of power

iteration for refinement. In addition, we consider linear iterative methods as alternative refinement procedures. We also consider the cascadic multigrid method in the geometric settings for elliptic eigenvalue problems and show its uniform convergence under certain assumptions. We explore the applicability of such an eigensolver to the graph partitioning problem and stress the simplicity of the implementation of our multilevel algorithm in comparison to other partitioning schemes. Numerical tests are presented for computing the Fiedler vector of several practical graphs, and numerical results show the efficiency and optimality of our proposed cascadic multigrid algorithm.

1.3 Adaptive Space-Time Multigrid for the Numerical Valuation of Options

We introduce a space-time multigrid method for the pricing of barrier options. In particular, we consider pricing the value of up-and-out barrier options, discretized by the method of lines, using the implicit Euler method. We implement a space-time multigrid method in which the domain in space and time is treated simultaneously. We consider an adaptive coarsening technique in which the choice of coarsening in space or time is dependent on the discrete problem's degree of ansitropy at each level. We perform local Fourier analysis to find a suitable choice of our ansitropy constant. In addition, we consider the numerical solution of the problem both in the standard space-time domain and in a transformed domain. Finally, we discuss the appeal of such an algorithm over other methods for parabolic partial differential equations.

Chapter 2

On the Preservation of Connectedness in Spectral Bisections

2.1 Introduction

We consider the set of connected, undirected multigraphs with no self-loops, which we will be denoted by $\mathcal{G}$. We will represent the set of graphs in $\mathcal{G}$ with $|V| = n$ by $\mathcal{G}_n$. A graph $G = (V, E)$, where V is the vertex set and E the edge set can be uniquely determined by its Laplacian (for a review of graph theory refer to [4, 5, 12]). We have the following defintion.

Definition 2.1.1. *Let $G = (V, E)$ be a graph. We define the Laplacian matrix of G, denoted $L(G) \in \mathbb{R}^{n^2}$, $n = |V|$, as follows:*

$$L(G)_{(i,j)} := \begin{cases} d_{v_i} & for \quad i = j \\ -w_{e_{ij}} & for \quad i \neq j \end{cases}$$

where d_{v_i} is the degree of vertex $v_i \in V$, and $w_{e_{ij}}$ the weight of edge $(i, j) \in E$

We note that $L(G)$ is self-adjoint and positive semi-definite. In addition, the sum of any row (and, also, any column) of L is zero. This implies that $\lambda = 0$ is an eigenvalue of L, with corresponding eigenvector $w = (1, ..., 1)^T$. Let us order the eigenvalues of $L(G)$ as follows: $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$, and denote by $w_1, w_2, ..., w_n$ the corresponding eigenvectors. We see that $w_1 = \alpha(1, 1, ..., 1)^T$. The eigenvalue and eigenvector pair λ_2, w_2 have special significance and, for this reason, are given special names.

Definition 2.1.2. *The algebraic connectivity of a graph G, denoted by $a(G)$, is defined to be the second eigenvalue of the corresponding Laplacian matrix, $L(G)$, with eigenvalues defined in increasing order $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. An eigenvector corresponding to the eigenvalue $a(G)$ is called a Fiedler vector of G.*

The term Fiedler vector comes from the mathematician Miroslav Fiedler, who proved many results regarding the significance of this vector. His work involving irreducible matrices and the Fiedler vector can be found in [13, 14]. The Fiedler vector of Graph Laplacians has proven to be a useful quantity, finding an application in graph partitioning, which has proven useful in data mining [3], very-large-scale integration (VLSI) design [38], and parallel computing [21].

For the remainder of the chapter, we consider the bisection problem exclusively. The goal of a two way partition (bisection) is to cut a given graph into two parts, where the size of each subgraph is roughly equal, while minimizing the number of edges that go between them. The key to computing such a partition in practice is to try to obtain one that is near optimal, while at the same time, only requiring a reasonable amount of computing time.

It turns out that the Fiedler vector of the Laplacian of a graph proves to be a useful tool for partitioning a graph into two parts. This can be seen by noting the connection between the Rayleigh quotient of L and an edge cut, and recalling that the Fiedler vector minimizes the Rayleigh quotient in the subspace $\{x|(x, \mathbf{1}) = 0\}$. For a more detailed derivation of the connection, refer to [29]. Bisections of graphs involving a Fiedler vector are called spectral bisections. Fiedler proved a result regarding the connectivity of such a spectral bisection.

Theorem 2.1.3 (Fiedler's Bisection Theorem)**.** *Let $G = (V, E) \in \mathcal{G}_n$ and let u be an eigenvector of the Laplacian $L(G)$ corresponding to the second smallest eigenvalue $a(G)$, the algebraic connecivity of G. Let $V_1 = \{i \in V | u_i \geq 0\}$, where u_i, $i \in \{1, 2, ..., n\}$, is the coordinate of u corresponding to the i-th vertex in V. Then the subgraph G_1 of G generated by G on the subset V_1 of V is connected.*

A proof of this fact can be found in [14]. From this theorem, we have the following corollary regarding the two components of a spectral bisection of a connected graph.

Corollary 2.1.4. *Let $G \in \mathcal{G}_n$ be a connected graph with vertices $1, 2, ..., n$. Let y be a Fiedler vector of G. If $y_i \neq 0$ for all $i \in \{1, 2, ...n\}$, then the set of all alternating edges, i.e. edges (i, k) for which $y_i y_k < 0$ forms a cut C of G such that both banks of G are connected.*

While the above corollary shows connectedness for a spectral bisection with $y_i \neq 0$ for all i, it says nothing about the case when there exists components with $y_i = 0$. We treat both cases, and show the existence of a Fiedler vector for any connected graph $G \in \mathcal{G}_n$ such that the spectral bisection preserves connectedness. In addition, we show the class of graphs for which all Fiedler vectors produce connected bisections to be larger than described above.

Theorem 2.1.5 (Generalized Bisection Theorem)**.** *Let $G \in \mathcal{G}_n$ be a graph with $V = \{1, 2, ..., n\}$. Let $U \cong \mathbb{R}^k$ be the eigenspace of $a(G)$, and $W \subset U$ be the set of elements $u \in U$ such that the subgraphs G_1, G_2 of G, generated by G on the subsets $V_1 = \{i \in V | u_i \geq 0\}$, $V_2 = \{i \in V | u_i < 0\}$, respectively, are connected. Then W has positive measure.*

This theorem shows that for a given connected graph, the set of Fiedler vectors for which a bisection produces two connected subgraphs has positive measure. We aim to prove this result.

2.2 Plan of Proof

The overall plan of our proof is as follows: coarsen the set of nodes for which the Fiedler vector has zero component (referred to as zero valuated vertices), treat the cases where the zero valuated vertex is and isn't an accumulation point separately, and then show that the subset W is non-empty. More explicitly, the general flow of the remainder of the chapter is as follows:

Step 1: Consider the case of star graphs S_k, $k \in \mathcal{N}$.

We treat the class of star graphs first, in order to gain intuition and give motivation for the general case. We will see strong parallels between this case and the general setting.

Step 2: Transform the graph into components, isolating nodes which are zero valuated for all $v \in U$ and coarsening them.

We consider the subset of $\mathbb{R}^n$ for which U lies. We consider the components for which $v_i = 0$ for all $v \in U$, and coarsen them to a single vertex. We show that the non-zero components of the Fiedler vector, as well as the overall structure of the graph, remain unchanged.

Step 3: Check whether the zero valuated vertex is an accumulation point and, if so, the number of components r its removal produces.

We consider whether the given node is a accumulation point, and if it is not, apply existing theory regarding accumulation points with respect to Fiedler vectors. Therefore, it remains to treat the case where the zero valuated vertex is an accumulation point.

Step 4: Show the existence of Fiedler vectors that produce connected spectral bisections for graphs with a zero valuated accumulation point.

We use the formulation of L that the accumulation point produces and, taking inspiration from the case of S_k, show the existence of connected spectral bisection producing Fiedler vectors.

2.3 Spectral Bisection of S_k

We begin by treating the case of the graph S_k. We do not assume any conditions on the edge weights of the graph. Rather, we only assume that the central node has zero-valuation for all $x \in \{x | Lx = a(G)x\}$. If this is not the case, then Corollary 2.1.4 applies. Without loss of generality, we will take our central node to be ordered as the first vertex. This gives us the following form for

our Laplacian:

$$L(S_k) = \begin{pmatrix} d_0 & -d_1 & -d_2 & \cdots & -d_k \\ -d_1 & d_1 & 0 & \cdots & 0 \\ -d_2 & 0 & d_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -d_k & 0 & \cdots & 0 & d_k \end{pmatrix}_{(k+1)\times(k+1)}$$

Taking $y = (0\ y_1\ ...\ y_k)^T$, $y \in \{x|Lx = a(G)x\}$, we see $Ly = a(G)y$. Therefore, we have the following relations:

$$\sum_{i=1}^{k} d_i\, y_i = 0$$

$$d_i\, y_i = a(G)\, y_i \quad \forall\, i \in \{1, 2, ..., k\}$$

This implies that $d_i = a(G)$ for $i = 1, 2, ..., k$. Renormalizing the Laplacian, $\hat{L} = \frac{1}{a(G)}L$, we have the following form:

$$\hat{L}(S_k) = \begin{pmatrix} k & -\mathbf{1}_k^T \\ -\mathbf{1}_k & I_k \end{pmatrix}$$

where $\mathbf{1}_k = (1, 1, ..., 1)^T \in \mathbb{R}^k$ and $I_k \in R^{k\times k}$ is the identity matrix. We see that U is the set of all $(k+1)$-dimensional vectors v such that $v_0 = 0$ and $\sum_{i=1}^{k} v_i = 0$. Therefore, $U \cong \mathbb{R}^{k-1}$. For the set $\{i|v_i < 0\}$ to be non-void and generate a connected graph, we must have $v_i < 0$ for exactly one index. Therefore, the set W consists of all normalized Fiedler vectors such that $v_j < 0$ for some j and $v_i \geq 0$ for all $i \neq j$. We see this set has positive measure. We have now shown Theorem 2.1.5 for the subset $\{S_k|k \in \mathbb{N}\} \subset \mathcal{G}$.

2.4 Transformation of General Graphs

We now treat the general case. Consider some graph $G \in \mathcal{G}_n$, and the set of its Fiedler vectors $U \subset \mathbb{R}^n$. Let $i_1, i_2, ..., i_j$ be the j components for which $v_{i_k} = 0$ for all $v \in U$ and $k \in \{1, 2, ..., j\}$. Let us reorder the indices such that the components $i_1, i_2, ..., i_j$ are transformed to the indices $1, 2, ..., j$. Representing our Laplacian in block form:

$$L = \begin{pmatrix} L_0 & -A^T \\ -A & \tilde{L} \end{pmatrix}_{n\times n}$$

where $L_0 \in \mathbb{R}^{j\times j}$, $A \in \mathbb{R}^{(n-j)\times j}$, $\tilde{L} \in \mathbb{R}^{(n-j)\times(n-j)}$. Our Fiedler vectors take the form $v = (0_{1\times j}\ \tilde{y}^T)^T$, where $\tilde{y} \in \mathbb{R}^{n-j}$. This gives us that $\tilde{L}\,\tilde{y} = a(G)\,\tilde{y}$ and $A^T\tilde{y} = 0$.

We aim to coarsen G, so that the vertices that make up L_0 become a single node, and both $\tilde{L}$

and $\tilde{y}$ remain unchanged. We introduce the interpolation matrix:

$$I_H^h = \begin{pmatrix} \mathbf{1}_j & 0_{j\times(n-j)} \\ 0_{n-j} & I_{n-j} \end{pmatrix}_{n\times(n-j+1)}$$

The Laplacian $(I_H^h)^T L I_H^h$ preserves $\tilde{L}$. In addition, $\tilde{y}$ is preserved under $(I_H^h)^T$. The coarsened Laplacian $\hat{L} = (I_H^h)^T L I_H^h \in \mathbb{R}^m$, $m = n - j + 1$, is represented by

$$\hat{L} = \begin{pmatrix} l_0 & -a^T \\ -a & \tilde{L} \end{pmatrix}_{m\times m}$$

From $\hat{L}(I_H^h)^T y = a(G)(I_H^h)^T y$, we obtain that $a = A\,\mathbf{1}_j$ and $l_0 = \mathbf{1}_j^T L_0\,\mathbf{1}_j$. What remains is to consider the properties of the M-matrix $\tilde{L}$.

2.5 Accumulation Points and Reducibility

We consider the reducibility properties of $\tilde{L}$. We begin by recalling the concept of reducibility. We have the following definitions, from [44].

Definition 2.5.1. *A matrix $A \in \mathbb{R}^{n\times n}$ is reducible if there exists a permutation matrix $P \in \mathbb{R}^{n\times n}$ such that*

$$PAP^T = \begin{pmatrix} A_{1,1} & A_{1,2} \\ 0 & A_{2,2} \end{pmatrix}$$

Definition 2.5.2. *A matrix $A \in \mathbb{R}^{n\times n}$ has degree of reducibility r if there exists a permutation matrix $P \in \mathbb{R}^{n\times n}$ such that*

$$PAP^T = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & \cdots & A_{1,r} \\ 0 & A_{2,2} & \cdots & \cdots & A_{2,r} \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & A_{r,r} \end{pmatrix}$$

with $A_{i,i}$ irreducible, $1 \le i \le r$.

We consider the reducibility properties of $\tilde{L}$. We note that $\tilde{L}$ is reducible if and only if the vertex v_0 is an accumulation point of G. If the removal of v_0 from G produces r connected components, then $\tilde{L}$ has degree of reducibility r. Assuming $\tilde{L}$ to already be in block reducible form, $\hat{L}$ has the

following representation:

$$\hat{L} = \begin{pmatrix} l_0 & -a_1^T & -a_2^T & \cdots & -a_r^T \\ -a_1 & L_1 & 0 & \cdots & 0 \\ -a_2 & 0 & L_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -a_r & 0 & \cdots & 0 & L_r \end{pmatrix}$$

Similarly, characteristic valuations can be put in the form $y = (0\ \tilde{y}^T) = (0\ y_1^T\ y_2^T\ ...\ y_r^T)^T$. The concept of reducibility is closely connected to the idea of connectedness. In particular, we have

Lemma 2.5.3. *A graph G is connected if and only if its corresponding Laplacian matrix $L(G)$ is irreducible.*

We begin by treating the case in which v_0 is not an accumulation point. This implies that $\tilde{L}$ is irreducible. We recall the following theorem, proven by Fiedler in [14].

Theorem 2.5.4. *Let A be an $n \times n$ non-negative, irreducible, symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. Let v be a vector such that, for fixed $s \geq 2$, $Av \geq \lambda_s v$. Then $M = \{i \mid v_i \geq 0\}$ is non-void, and the degree of reducibility of $A(M)$ is less than or equal to $s - 2$.*

Let $0 = \lambda_1 < \lambda_2 \leq ... \leq \lambda_m$ and $\mu_1 \leq \mu_2 \leq ... \leq \mu_{m-1}$ be the eigenvalues of $\hat{L}$ and $\tilde{L}$, respectively. We see $\lambda_1 = 0$ and $\lambda_1 < \lambda_2$ from the properties of the Laplacian of a connected graph. We do not have the same properties for $\tilde{L}$, because $\tilde{L}$ is not a Laplacian, but rather a M-matrix.

Let v be an eigenvector of $\tilde{L}$ corresponding to μ_2. Consider the Gershgorin bound of $\tilde{L}$, given by $g = max_{1 \leq i \leq m-1} |\tilde{L}(i,i)|$. We have that the matrix $B = gI - \tilde{L}$ is non-negative, with eigenvalues $g - \mu_1 \geq g - \mu_2 \geq ... \geq g - \mu_{m-1} > 0$. Therefore, by Theorem 2.5.4, we see that the set $\{i | v_i \geq 0\}$ is non-void and irreducible. We aim to show that $\tilde{y}$ is an eigenvector of μ_2. We begin by recalling the eigenvalue interlacing theorem.

Theorem 2.5.5. *Let S be a real $n \times m$ matrix such that $S^T S = I$ and let A be a symmetric $n \times n$ matrix with eigenvalues $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. Define $B = S^T A S$ and let B have eigenvalues $\mu_1 \leq \mu_2 \leq ... \leq \mu_m$. Then the eigenvalues of B interlace those of A, namely*

$$\lambda_{n-m+i} \geq \mu_i \geq \lambda_i$$

From this we see that $0 = \lambda_1 \leq \mu_1 \leq a(G) \leq \mu_2$. From $\tilde{L}\tilde{y} = a(G)\tilde{y}$, we see that $a(G)$ is an eigenvalue and $\tilde{y}$ an eigenvector of $\tilde{L}$. Therefore, $\mu_1 = a(G)$ or $\mu_2 = a(G)$, or both. We recall that the eigenvector of μ_1 is either non-negative, or non-positive, by the Courant-Fisher Theorem. Because $\tilde{y}$ contains both positive and negative values, we know $\tilde{y}$ must be an eigenvalue of $\mu_2 = a(G)$.

By Theorem 2.5.4, we see that the matrix generated by $\{i | \tilde{y}_i \geq 0\}$ is non-void and irreducible. However, we have that $\tilde{y}_i \neq 0$ for all i. Therefore, both $\{i | \tilde{y}_i > 0\}$ and $\{i | \tilde{y}_i < 0\}$ are non-void and irreducible. This implies that the graphs generated by $\{i | \tilde{y}_i > 0\}$ and $\{i | \tilde{y}_i < 0\}$ are non-empty and

connected. Therefore, on the larger graph, represented by $\hat{L}$, we have that the graphs generated by $\{i|y_i \geq 0\}$ and $\{i|y_i < 0\}$ are non-empty and connected. This completes the proof of the case when v_0 is not an accumulation point. We present this in the form of a theorem.

Theorem 2.5.6. *Let $G \in \mathcal{G}_n$ be a connected graph with vertices $1, 2, ..., n$. Let y be a Fiedler vector of G. If G is still connected upon the removal of vertices such that $y_i = 0$, then the set of all alternating edges, i.e. edges (i, k) for which $y_i y_k < 0$ forms a cut C of G such that both banks of G are connected.*

We see that this extends the results of Fiedler to a more general class of graphs. It suffices to consider only the k indices for which $y_i = 0$ for all $y \in U$, because the set of Fiedler vectors for which $y_i = 0$ for more than k indices has measure zero. All that remains is to treat the case where v_0 is an accumulation point of G.

2.6 Measure of W for Graphs with Zero Valuated Accumulation Sets

We now consider the case where v_0 is an accumulation point of G, and aim to show that for this case W has positive measure. We recall the following theorem with respect to accumulation points, from [14].

Theorem 2.6.1. *Let G be a connected graph, and y a characteristic valuation of G. Let k be a point of articulation of G and let $G_0, G_1, ..., G_r$ be the components of the graph obtained from G by removing the vertex k and all adjacent edges. Then:*

(i) If $y_k > 0$, then exactly one of the components G_i contains a vertex negatively valuated in y. For all vertices s in the remaining components $y_s > y_k$.

(ii) If $y_k = 0$ and there is a component G_i containing both positively and negatively valuated vertices, then there is exactly one such component, all remaining being zero valuated.

(iii) If $y_k = 0$ and no component contains both positively and negatively valuated vertices, then each component G_i contains either only positively valuated, or negatively valuated, or only zero valuated vertices.

We see that for our purposes, we only consider case (iii). We can assume our accumulation point v_0 to be the only zero valuated point, for the set of Fiedler vectors which have more than one zero-accumulation point after our coarsening procedure have measure zero. Therefore, case (ii)

does not apply. With $\hat{L}$ in the form

$$\hat{L} = \begin{pmatrix} l_0 & -a_1^T & -a_2^T & \cdots & -a_r^T \\ -a_1 & L_1 & 0 & \cdots & 0 \\ -a_2 & 0 & L_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -a_r & 0 & \cdots & 0 & L_r \end{pmatrix}$$

we see that for each L_i, we have that the component of y corresponding to L_i, denoted y_i, is either positive or negative, i.e. $y_i > 0$ or $y_i < 0$. Therefore, we see that a valuation $y \in W$ if $y_j < 0$ for some j and $y_i > 0$ for all $i \neq j$.

This is very similar to the case of star graphs $\{S_r | r \in \mathbb{N}\}$. However, we are working with vectors y_i, rather than numbers. However, by a renormalization, we can treat it as an equivalent case. Suppose we renormalize each y_i so that $a_i^T y_i = 1$. Then we consider the Fiedler vector $y = (0\ \alpha_1 y_1^T\ \alpha_2 y_2^T\ ...\ \alpha_r y_r^T)^T$, $\alpha_1, \alpha_2, ..., \alpha_r \in \mathbb{R}$.

What remains to be shown is that y must take this form, for fixed y_i. This can be shown by proving the eigenvalue $a(G)$ has multiplicity one for $L_1, L_2, ..., L_r$. Rather than show this, we recall that the space of eigenvectors which are either positive or negative has dimension one, by the Perron-Frobenius Theorem, since $L_1, L_2, ..., L_r$ are all M-matrices. Therefore, by Theorem 2.6.1, it follows that y must be in the form $y = (0\ \alpha_1 y_1^T\ \alpha_2 y_2^T\ ...\ \alpha_r y_r^T)^T$.

We see that having $y \in W$ is equivalent to $\alpha_j < 0$ for some j, and $\alpha_i > 0$ for all $i \neq j$. This is exactly the case we were treating for the star graph S_r. It follows that W has positive measure. Theorem 2.1.5 is proven.

2.7 Conclusion

We have proven more general results with respect to the characterization of connected spectral bisection producing valuations. We extended the results of Fiedler to a larger class of graphs, and characterized the cases where valuations produce connected graphs. We showed the set of Fiedler vectors which produce connected spectral bisections to have positive measure. We hope this fosters further research into the field of spectral partitioning.

Chapter 3

A Cascadic Multigrid Algorithm for Computing the Fiedler Vector of Graph Laplacians

3.1 Introduction

Computation of the Fiedler vector of Graph Laplacians has proven to be a relevant topic, and has found applications in areas such as graph partitioning and graph drawing [29]. We will focus on graph partitioning as an application. Graph partitioning is an interesting and increasingly relevant area in mathematics. This problem has numerous uses, including data mining [3], very-large-scale integration (VLSI) design [38], and parallel computing [21]. Suppose there is a program that needs to be run simultaneously on k processors. The goal is to divide the computational tasks between the processors evenly, while at the same time, minimizing the overhead communication. This can be easily converted to a k-way partitioning problem, with the vertices being the tasks, the edges the connections between them, and the partitions the processors [21]. The goal of a k-way partitioning problem is to partition a given graph into k parts, where the size of each partition is roughly equal, while minimizing the number of edges that go between partitions. We will formalize this concept later. The key to computing such a partition in practice is to obtain one that is nearly optimal, while at the same time, only requiring a reasonable amount of computing time.

It turns out that the Fiedler vector of the Laplacian of a graph proves to be a good tool for partitioning a graph into two parts. In Section 3.2.2 we will show the connection between the Fiedler vector and graph partitioning. There have been a number of techniques implemented for computation of the Fiedler vector for the purpose of partitioning, most notably by Barnard and Simon [2]. They implemented a multilevel coarsening procedure, using maximal independent sets and created a matching from them. For the refinement procedure, Rayleigh quotient iteration was

used. We note that the term refinement refers to the smoothing process that occurs, and has a different meaning in the multigrid literature. Although at the time this was significantly faster than the standard recursive spectral bisection, it leaves room for improvement. The majority of the improvement has been in the form of coarsening algorithms. Better coarsening techniques, such as heavy edge matching (HEM), have been used more frequently, and have exhibited much shorter runtimes [26, 27]. However, very little has been done in the improvement of refinement procedures.

In this chapter, we introduce a new and fast coarsening algorithm, based on the conecpt of heavy edge matching, to create a more rapid coarsening procedure. For refinement, we implement a form of power iteraiton, and also consider linear iterative methods for refinement as well. For both our coarsening and refinement procedures we have created algorithms that are easily implemented. While heavy edge matching is complicated and tough to implement in high level programming languages, since it involves selecting an edge with heaviest weight between two unmatched vertices, heavy edge coarsening is significantly easier because we do not need to worry about whether the vertex has been aggregated or not. For the refinement procedure, power iteration does not require the inversion of a matrix, making its use much more straightforward than for Rayleigh quotient iteration, which requires some technique to approximately invert the matrix. Based on these two improved components, we propose a cascadic multigrid (CMG) method to computing the Fiedler vector. This is a purely algebraic approach which only uses the given graph. Moreover, although the purely algebraic approach is technically difficult to analyze, we consider the CMG method for the elliptic eigenvalue problem in the geometric setting. Based on the standard smoothing property and approximation property, we show that the geometric CMG method converges uniformly for the model problem, which indirectly provides theoretical justification of the efficiency of the CMG method. This also shows the potential of our CMG method for solving other eigenvalue problems from different applications.

The rest of the chapter is organized as follows. We begin by defining the Laplacian of a graph and the Fiedler vector in Section 3.2. We introduce the concept of a partition and build the connection between the Fiedler vector and partitioning. In Section 3.3, we introduce our cascadic algorithm. Details of our coarsening and our refinement procedures are introduced in Section 3.4 and 3.5 respectively. In Section 3.6, we consider our CMG method for elliptic eigenvalue problem in geometric settings, and prove its uniform convergence. Finally, we perform numerical tests on our algorithm to compare it to existing methods in Section 3.7.

3.2 Preliminary

In this section, we introduce the basic concepts, notation, and properties of the graph Laplacian, Fiedler vector, and graph partitioning.

3.2.1 Graph Laplacian and Fiedler Vector

We begin by formally introducing the concept of a graph Laplacian and Fiedler vector. We start with the concept of a graph. A graph $G = (V, E)$ is said to be undirected if the edges have no orientation. A graph is a multigraph if $e_{(i,i)} \notin E$ for all $1 \leq i \leq |V|$ ($|V|$ is the number of vertices). For the remainder of this chapter, we assume that all graphs are undirected and multigraphs.

We consider the task of representing a graph in matrix form. One of the most natural representations is through its Laplacian. The Laplacian of a graph is defined as follows:

Definition 3.2.1. *Let $G = (V, E)$ be a graph. We define the Laplacian matrix of G, denoted $L(G) \in \mathbb{R}^{n \times n}$ (or just L for short), $n = |V|$, as follows:*

$$L(G)_{(i,j)} := \begin{cases} d_{v_i}, & \text{for} \quad i = j, \\ -w_{e_{(i,j)}}, & \text{for} \quad i \neq j, \end{cases}$$

where d_{v_i} is the degree of v_i, and $w_{e_{(i,j)}}$ is the weight of the edge connecting v_i and v_j.

The Laplacian $L(G)$ is self-adjoint, positive semi-definite, and diagonally dominant. In addition, the sum of any row (and also, any column) of L is zero. Therefore $\lambda = 0$ is an eigenvalue of L, with corresponding eigenvector $\mathbf{1} = (1, ..., 1)^T$. Let us order the eigenvalues of $L(G)$ as follows: $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$, and denote by $w_1, w_2, ..., w_n$ the corresponding eigenvectors. We have already seen that $w_1 = \alpha\mathbf{1}$. We now consider λ_2 and w_2. This eigenvalue and eigenvector pair has special significance and, for this reason, are given special names.

Definition 3.2.2. *The algebraic connectivity of a graph G, denoted by $a(G)$, is defined to be the second smallest eigenvalue of the corresponding Laplacian matrix $L(G)$, with eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ and eigenvectors $w_1, w_2, ..., w_n$. The eigenvector w_2, corresponding to the eigenvalue $a(G)$, is called the Fiedler vector of G.*

The term Fiedler vector comes from the mathematician Miroslav Fiedler, who proved many results regarding the significance of this eigenvector. His work involving irreducible matrices and the Fiedler vector can be found in [13, 14].

3.2.2 Graph Partitioning

Based on the graph Laplacian and Fiedler vector, we consider the graph partitioning problem and put it in an optimization framework. This section closely follows the work done in [22, 29, 37], and is mainly expository in nature. We begin by introducing basic terminology.

Definition 3.2.3. *Let $G = (V, E)$ be a graph, and $\mathcal{P} = \{V_1, V_2, ..., V_k\}$ be a partition of V. Assume that $|V| = km$, for some $m \in \mathbb{Z}_+$. We say that the partition $\mathcal{P}$ is optimal if $|V_1| = |V_2| = ... = |V_k|$ and the edge cut of $\mathcal{P}$ is minimized, where $e(\mathcal{P})$ is given by*

$$e(\mathcal{P}) := \sum_{i \in V_r, j \in V_s, r \neq s} w_{e_{(i,j)}}.$$

Although this definition is for a general k-way partition, for the remainder of this section we will restrict ourselves to the two-way partitioning problem. We consider the Rayleigh quotient of a Laplacian matrix L. We have the following lemma, from [37], regarding its representation.

Lemma 3.2.4. *Let the $n \times n$ matrix L be the Laplacian of some graph G. Then the Rayleigh quotient of L is given by*

$$r_L(x) = \frac{1}{\|x\|_2} \sum_{1 \le i < j \le n} w_{e_{(i,j)}} (x_i - x_j)^2.$$

This property proves useful when considering partitions. Consider some partition $\mathcal{P} = \{V_1, V_2\}$. Let us define a vector $x^* \in \mathbb{R}^n$ as follows:

$$x_i^* = \begin{cases} +\frac{1}{\sqrt{n}}, & \text{if} \quad i \in V_1, \\ -\frac{1}{\sqrt{n}}, & \text{if} \quad i \in V_2. \end{cases}$$

For this specific choice x^*, the Rayleigh quotient gives us

$$r_L(x^*) = \frac{4}{n} \sum_{i \in V_1, j \in V_2} w_{e_{(i,j)}}.$$

Note the similarity to the definition of an edge cut. For a given bisection $\mathcal{P}$, we have $r_L(x) = \frac{n}{4} e(\mathcal{P})$. This implies that the problem of finding the partition $\mathcal{P}$ that minimizes the edge cut is equivalent to the problem of finding the x^*, with $x_i^* = \pm\frac{1}{\sqrt{n}}$, that minimizes $r_L(x^*)$.

We can present the problem of finding the optimal two-way partition in this framework as well. The condition that $|V_1| = |V_2|$ is equivalent to requiring that $\mathbf{1}^T x^* = 0$. It now follows that the problem of finding the optimal two-way partition $\mathcal{P}$ of a graph G can be viewed as the following optimization problem:

$$\text{minimize } r_L(x^*), \text{ subject to } x_i^* = \pm\frac{1}{\sqrt{n}} \text{ and } w_1^T x^* = 0.$$

The solution of this problem is hard to compute, but if we relax the condition of $x_i^* = \pm\frac{1}{\sqrt{n}}$ to $\|x^*\|_2 = 1$, we can find a solution using Lagrange multipliers. We obtain

$$\nabla f(x^*) - \mu_1 \nabla g_1(x^*) - \mu_2 \nabla g_2(x^*) = 2Lx^* - 2\mu_1 x^* - \mu_2 w_1 = 0.$$

Premultiplying by w_1^T gives us

$$2w_1^T L x^* - \mu_2 = 2(Lw_1)^T x^* - \mu_2 = -\mu_2 = 0 \Rightarrow Lx^* = \mu_1 x^*.$$

The optimal x^* is an eigenvector of L. By the Courant-Fisher Theorem, it follows that the Fiedler vector $y = w_2$ is the minimizer. It is this eigenvector that we will use for our bisection. It is in this way that our eigensolver for the Fiedler vector becomes of great use in graph partitioning. Graph partitioning methods that use computation of the Fiedler vector are referred to as spectral methods. Now that we have established the connection between graph partitioning and the Fiedler

vector, we will now discuss our cascadic algorithm for computing this vector.

3.3 Cascadic MG Method for Computing the Fiedler Vector

In this section, we introduce our cascadic MG (CMG) algorithm for computing the Fielder vector. Our CMG algorithm is a purely algebraic approach, and the multilevel structure is constructed from the graph directly. Therefore, similar to standard algebraic MG (AMG) method, the new algorithm consists of three steps: a setup phase, a solving phase on the coarsest level, and a cascadic solving phase (also called refinement phase in our chapter). The process works as follows:

Step 1: Coarsen our graph G_0 iteratively to subgraphs $G_1, G_2, ..., G_J$.

Taking inspiration from AMG coarsening and graph matching, we introduce a technique we call heavy edge coarsening (HEC). This coarsening procedure produces a restriction matrix I_i^{i+1} at each level i, and its transport will be used as the prolongation. This creates a multilevel structure of coarse Laplacians $L_0, L_1, ..., L_J$. In general, the coarsening phase of a multilevel algorithm of this form tends to be the most expensive part of the procedure.

Step 2: Solve for the Fiedler vector on the coarse graph G_J.

In practice the coarse graph tends to be small in size (usually $|V| < 100$). The technique implemented on this level is not extremely relevant for single computations of the vector. However, for applications which may require this eigenvalue computation a large number of times (such as recursive spectral bisection, for large k), this becomes more of a relevant issue. The commonly used eigensolver on this coarse level, in the absence of a good intial guess, is the Lanczos algorithm. However, in our implementation we over-coarsen to $|V| < 25$ and use power iteration on a random vector, sampled from a Gaussian distribution.

Step 3: Prolongate the Fiedler vector from coarse graph G_J back to G_{J-1} and apply several steps of refinement (smoothing). Repeat this step until the finest level.

We map the Fiedler vector to the finer graph at each level and perform some form of refinement (smoothing). We mainly consider a variant of power iteration (PI), but also introduce other smoothers commonly used in multigrid cycles. Because our focus here is the Fiedler vector, we need to keep the iterators orthogonal to the constant vector.

Traditionally, in the MG method literature, steps 2 and 3 together are called the CMG method. However, in non-spectral methods for graph partitioning, this is not the case, and for this reason we maintain the three-step structure that is prevalent in the literature. We present the core of our cascadic eigensolver in Algorithm 1.

Algorithm 1: Multilevel Cascadic Eigensolver

Input: graph Laplacian matrix $L^0 \in \mathbb{R}^{n_0 \times n_0}$

Output: approximate Fiedler vector $\tilde{y}^{(0)}$

Step 1: Setup Phase

set $i = 0$

while $n_i > 25$ **do**

  $I_i^{i+1} \leftarrow \text{HEC}(L^i)$

  $L^{i+1} = I_i^{i+1} L^i (I_i^{i+1})^T$

  $i = i + 1$

$J \leftarrow i$

Step 2: Coarsest Level Solving Phase

$\tilde{y}^{(J)} \leftarrow \text{PI}(L^J, randn(n_J))$

Step 3: Cascadic Refinement Phase

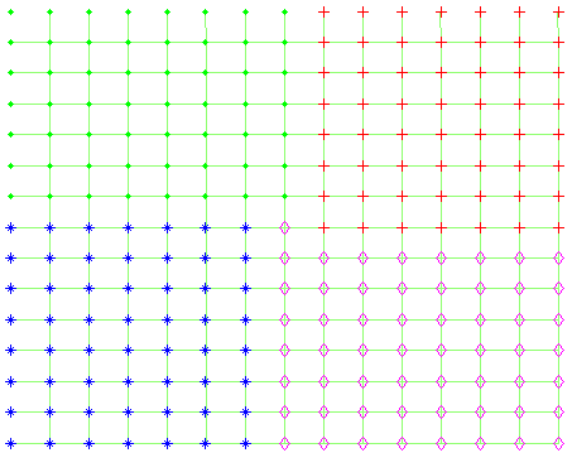
for *j=J-1,J-2,...,0* **do**

  $\hat{y}^{(j)} = (I_i^{i+1})^T \tilde{y}^{(j+1)}$

  $\tilde{y}^{(j)} \leftarrow \text{PI}(L^j, \hat{y}^{(j)})$

As an initial illustration of our algorithm, we give an example of our partitioning algorithm implemented on a structured grid of size 15×15. We consider a 4-way partition and observe the partitions that we obtain. The image of this partition can be found in Figure 3.1. The four partitions are all connected and almost exactly the same size, as expected.

Figure 3.1. Four-Way Partition of a Structured 15×15 Grid Using the Multilevel Cascadic Eigensolver



As mentioned before, because the size of the coarsest graph is very small, and power iteration is efficient, our focus is on the first and third steps. We will first introduce the heavy edge coarsening scheme we proposed for the setup phase in the next section, and then present our cascadic refinement

scheme in detail in Section 3.5.

3.4 Heavy Edge Coarsening

In this section, we will discuss the coarsening algorithm used for the setup phase. The goal for this step is to coarsen a graph quickly, while also maintaining some semblance of its structure. In practice, the coarsening procedure tends to dominate the run time of the multilevel eigensolver in general. There is a great trade-off between fast and optimal coarsening techniques. By following some of the themes currently in place in the existing literature [29, 26, 27], we propose a new coarsening algorithm, which takes a stance somewhere in between. In order to introduce our coarsening algorithm, we begin by considering matching as a coarsening technique. The concept of a matching formally is as follows:

Definition 3.4.1. *Let $G = (V, E)$. A matching is a subset $E^* \subset E$, such that no two elements of E^* are incident on the same vertex. A matching E^* is said to be a maximal matching if there does not exist an edge $e_{i,j} \in E \backslash E^*$ such that $E^* \cup \{e_{i,j}\}$ is still a matching.*

For our purposes, the matching computed at each level is always a maximal matching. A matching is computed at each level, and the edges in the matching are collapsed to form the coarser graph. We consider the class of matching algorithms concerned with finding the matching with the heaviest edge weight. A matching of heavy edges would make an ideal coarse graph for our multilevel eigenproblem. The reason for this is related to graph partitioning. This coarsening procedure creates a smaller edge cut on coarse levels for partitions, which results in smaller edge cuts for the finer graphs. Even though we are not refining partitions, this concept still applies, due to the close connection between the Fiedler vector and graph partitioning. To do a matching of heavy edges optimally is rather expensive because it would require searching for the heaviest weighed edge incident to two unmatched vertices at each step. In practice, the vertices are usually visited in a random order, and the heaviest weighed incident edge with an unmatched vertex is chosen. Such a technique produces a somewhat less optimal partition, but is much faster in practice. We adopt this procedure in our coarsening algorithm.

Moreover, we choose to perform a more aggressive coarsening procedure, rather than matching, because it results in less coarse levels in the multilevel scheme. In addition, when considering heavy edge schemes, the aggressive coarsening procedure is significantly easier to implement than its matching counterpart because we consider mapping each node to the incident vertex with the heaviest edge, rather than picking the heaviest edge with an unmatched vertex. Moreover, rather than mapping every vertex, we only perform this mapping procedure to vertices that have yet to be mapped to. In general, this will not result in a matching. Therefore, we call this technique heavy edge coarsening (HEC). This procedure can easily be implemented. For the specific details regarding its implementation, we refer to Algorithm 2.

The HEC procedure proves to be a fast and efficient means of coarsening. The structure of the finer graph is well represented, making the refinement process of power iteration converge quickly.

In addition, one of the biggest benefits of HEC is the relatively small number of coarse levels m required. We will introduce this concept formally, in the form of a lemma.

Lemma 3.4.2. *Let $G_i = (V_i, E_i)$ be a connected graph. Let $I_i^{i+1}[HEC] \in \mathbb{R}^{n_{i+1}^{HEC} \times n_i}$ be a restriction matrix defined by heavy edge coarsening and $I_i^{i+1}[E_i^*] \in \mathbb{R}^{n_{i+1}^{M} \times n_i}$ be a restriction matrix defined by a matching E_i^*. Define $k_{HEC}^i = n_{i+1}^{HEC}/n_i$ and $k_M^i = n_{i+1}^M/n_i$ Then we have $1/n_i \leq k_{HEC}^i \leq 0.5$ and $0.5 \leq k_M^i \leq 1$.*

Proof. From the definition of a matching, we have that n_{i+1}^M cannot be less than half of n_i. For the bounds on k_{HEC}^i, we note that for our HEC algorithm, every node in V_i is mapped to another node, or has been mapped to. This implies that n_{i+1}^{HEC} is at most half of n_i. The lower bound results from taking a HEC procedure on a graph G_i such that G_{i+1} is a single node. □

Algorithm 2: Heavy Edge Coarsening (HEC)

Input: graph Laplacian matrix $L^i \in \mathbb{R}^{n_i \times n_i}$
Output: restriction matrix I_i^{i+1}

```
c ← 0
p ← randperm(n_i)
q ← zeros(n_i, 1)
for i = 1, 2, ..., n_i do
    if q(p(i)) = 0 then
        m ← argmin(L(:, p(i)))
        if q(m) = 0 then
            c ← c + 1
            q(m) = c
            q(p(i)) = c
        else
            q(p(i)) = q(m)
I_i^{i+1} ← zeros(c, n_i)
for i = 1, 2, ..., n_i do
    I_i^{i+1}(q(i), i) = 1
```

We have given a bound for the value of k_{HEC} (we drop the superscript i for simplicity). Given below in Table 3.1 are samples of what values k_{HEC} takes in practice for different graphs. As expected, the values taken in practice are significantly below the given bound of 0.5. The details about those graphs and more numerical results regarding k_{HEC} are given in Section 3.7.

Table 3.1. Sample values of k_{HEC}

Graph	Sample k_{HEC}^0 Value
144	0.1893
598a	0.2024
auto	0.1742

What remains to be explored is the properties of the restiction matrix I_i^{i+1}. The most important fact that we require is that the coarse matrix created by the restriction matrix is still a Laplacian matrix of the coarse graph. In addition, we want to inspect whether or not the constant eigenvector $\mathbf{1} = (1, ..., 1)^T$ is preserved under restrictions and prolongations. We also consider issues of orthogonal solutions with respect to the refinement procedure. Those properties are summarized in the following proposition.

Proposition 3.4.3. *Let $I_i^{i+1} \in \mathbb{R}^{n_{i+1} \times n_i}$ be a restriction matrix defined by HEC. Then we have the following:*

1. *$(I_i^{i+1})^T \mathbf{1}^{i+1} = \mathbf{1}^i$. That is, the eigenvector $\mathbf{1}$ is preserved under refinement.*
2. *If L^i is a Laplacian matrix, then $L^{i+1} = I_i^{i+1} L^i (I_i^{i+1})^T$ is also a Lapacian matrix. In particular, $L^{i+1}\mathbf{1}^{i+1} = 0$.*
3. *Let $u \in \mathbf{1}^\perp = \{u | (u, \mathbf{1}) = 0\} \subset \mathbb{R}^{n_i}$. Then $I_i^{i+1} u \in \mathbf{1}^\perp \subset \mathbb{R}^{n_{i+1}}$. However, in general, $(I_{i-1}^i)^T u \notin \mathbf{1}^\perp \subset \mathbb{R}^{n_{i-1}}$.*

Proof. We begin with (1). This follows from the fact that each vertex in V_i is mapped to only one vertex in V_{i+1}. However, $I_i^{i+1}\mathbf{1}^i \neq \mathbf{1}^{i+1}$. This is expected, as the number of vertices in V_i mapped to a given vertex $v_j \in V_{i+1}$ varies.

To prove (2), we need to show that L^{i+1} is still symmetric, with positive diagonal and nonpositive offdiagonal, with $L^{i+1}\mathbf{1}^{i+1} = 0$. We begin by decomposing L^i into its degree matrix D^i and adjacency matrix A^i. This gives us $L^{i+1} = I_i^{i+1} D^i (I_i^{i+1})^T - I_i^{i+1} A^i (I_i^{i+1})^T$. $I_i^{i+1} D^i (I_i^{i+1})^T$ is still a degree matrix, and $I_i^{i+1} A^i (I_i^{i+1})^T$ an adjacency matrix. We show that L^{i+1} is a Laplacian by taking $L^{i+1}\mathbf{1}^{i+1} = I_i^{i+1} L^i (I_i^{i+1})^T \mathbf{1}^{i+1} = I_i^{i+1} L^i \mathbf{1}^i = 0$.

Part (3) of the Proposition can be shown as follows. Let $u \in \mathbf{1}^\perp \subset \mathbb{R}^{n_i}$. We have $(I_i^{i+1} u, \mathbf{1}^{i+1}) = (u, (I_i^{i+1})^T \mathbf{1}^{i+1}) = (u, \mathbf{1}^i) = 0$. Therefore, $I_i^{i+1} u \in \mathbf{1}^\perp \subset \mathbb{R}^{n_{i+1}}$. Looking at $(I_{i-1}^i)^T u$, we see $((I_{i-1}^i)^T u, \mathbf{1}^{i-1}) = (u, I_{i-1}^i \mathbf{1}^{i-1}) \neq (u, \mathbf{1}^i) = 0$, since $I_{i-1}^i \mathbf{1}^{i-1} \neq \mathbf{1}^i$. □

3.5 Refinement Strategies

Given an approximate Fiedler vector $y^{(i+1)}$ on a coarse graph G_{i+1}, we aim to find an optimal manner to project this vector back to the finer graph G_i and refine it to an approximate Fiedler vector $y^{(i)}$ on G_i. We begin by considering the projection problem. The most natural way to project $y^{(i+1)}$ to G_i is to use the restriction matrix I_i^{i+1} obtained from coarsening, define our prolongation matrix to be $(I_i^{i+1})^T$, and let the initial approximation be $\tilde{y}^{(i)} = (I_i^{i+1})^T y^{(i+1)}$. However, we have to concern ourselves with orthogonality to the eigenvector $\mathbf{1}$. From proposition 4.44, we have that $(\tilde{y}^{(i)}, \mathbf{1}^i) \neq 0$. Therefore, before we can perform any sort of eigenvalue refinement procedure, we require our inital vector to be in the subspace $\mathbf{1}^\perp = \{u | (u, \mathbf{1}) = 0\}$. This can be accomplished by one iteration of Gram-Schmidt. From here, the orthogonality will be approximately maintained, since $\mathbf{1}^\perp$ is L-invariant.

Given an approximation $\tilde{y}^{(i)}$, we can refine it in a number of ways. We mainly consider power iteration as a refinement scheme in our CMG algorithm because of its simplicity. In this way we take advantage of the sparsity of our Laplacian. We also will introduce other alternative approaches in this section.

3.5.1 Power Iteration

Because the Fiedler vector corresponds to the second smallest eigenvalue of the graph Laplacian, we cannot apply the power iteration directly. Therefore, we compute a Gershgorin bound on the eigenvalues of a Laplacian L by taking $g = \max_i \sum_{1 \leq j \leq n} |l_{i,j}|$, the greatest row sum of the absolute value of the entries of L. From the Gershgorin circle Theorem, we have that all the eigenvalues of $gI - L$ are strictly positive, and are ordered oppositely, with eigenvalues $g - \lambda_1, g - \lambda_2, ..., g - \lambda_n$. The eigenvectors remain unchanged. In this way it suffices to perform power iteration on $gI - L$, coupled with an intial orthogonalization to $w_1 = \mathbf{1}$. We note that $\mathbf{1}^{\perp}$ is also invariant under $gI - L$. This variant of power iteration is detailed in Algorithm 3.

We proceed by examining the stopping criterion for power iteration, and the number of iterations required for convergence at a given level. Let u^0 denote our initial guess, and u^k represent the normalized vector resulting from k iterations. For our algorithm, the stopping criterion is given by $(u^k, u^{k-1}) > 1 - \delta$, for some given tolerance δ. We note that this is equivalent to $||u^k - u^{k-1}||^2 < 2\delta$. We recall the folowing result, with respect to power iteration on an arbitrary symmetric matrix.

Theorem 3.5.1. *Let A be a symmetric matrix with eigenvalues $\lambda_1 > \lambda_2 \geq ... \geq \lambda_n \geq 0$ and corresponding eigenvectors $w_1, w_2, ..., w_n$. Then power iteration, with intial guess u^0, $(u^0, w_1) \neq 0$, has convergence rate given by*

$$\sin \angle(u^k, w_1) < \left|\frac{\lambda_2}{\lambda_1}\right|^k \tan \angle(u^0, w_1).$$

A proof of this result can be found in [17]. For notational convenience, for the remainder of the chapter we will omit the $\angle$ sign when referring to the trigonometric function of an angle between two vectors. We are now ready to prove a result for our stopping criterion $||u^k - u^{k-1}||^2 < 2\delta$.

Algorithm 3: Power Iteration (PI)

Input: graph Laplacian matrix $L \in \mathbb{R}^{n \times n}$, initial guess $\tilde{y}^0$

Output: approximate Fiedler vector $\tilde{y}$

$g = max_i \sum_{1 \leq j \leq n} |l_{i,j}|$
$B_g = gI - L$
$u = \tilde{y}^0 - \frac{\mathbf{1}^T \tilde{y}^0}{n} \tilde{y}^0$
$u \leftarrow \frac{u}{\|u\|}$
$v \leftarrow zeros(n, 1)$
while $u^T v < 1 - tol$ **do**
    $v \leftarrow u$
    $u = B_g v$
    $u \leftarrow \frac{u}{\|u\|}$
$\tilde{y} = u$

Theorem 3.5.2. *Let $||u^k - w_2||^2 < 2\delta$ be the stopping criterion for power iteration with an intial guess u^0 for the matrix $B = gI - L$ where L is a Laplacian, with eigenvalues $g > g - \lambda_2 \geq g - \lambda_3 \geq ... \geq g - \lambda_n$ and corresponding normalized eigenvectors $w_1, w_2, ..., w_n$. Let $\alpha_i = (u^0, w_i)$ ($\alpha_1 = 0$, by assumption). Then the number of iterations required is bounded above by*

$$k_0 = \Big| \frac{C}{\lambda_3 - \lambda_2} \log \big(\frac{\tan^2(u^0, w_2)}{\delta} \big) \Big|_+ \tag{3.1}$$

where $| \cdot |_+ := \max(\cdot, 0)$ and $C < \frac{g - \lambda_2}{2}$.

Proof. Taking the result of Theorem 3.5.1, and applying it to our case, we obtain

$$\sin(u^k, w_2) < \big(\frac{g - \lambda_3}{g - \lambda_2} \big)^k \tan(u^0, w_2)$$

. Noting that our u^k is normalized, we have $\sin(u^k, w_2) = \sqrt{1 - (u^k, w_2)^2}$. The criterion $||u^k - w_2||^2 < 2\delta$ is equivalent to $(u^k, w_2) < 1 - \delta$. This is also equivalent to requiring that $\sqrt{1 - (u^k, w_2)^2} < \sqrt{\delta(2 - \delta)}$. We therefore aim to find the smallest k such that

$$\big(\frac{g - \lambda_3}{g - \lambda_2} \big)^k \tan(u^0, w_2) < \sqrt{\delta(2 - \delta)}$$

. Taking the logarithm of both sides and rearranging terms, we obtain

$$k > \frac{\log \big(\frac{\sqrt{\delta(2-\delta)}}{\tan(u^0, w_2)} \big)}{\log \big(\frac{g - \lambda_3}{g - \lambda_2} \big)}$$

. Taylor expanding the denominator about one, and moving terms, we have

$$k > \frac{g-\lambda_2}{2} \frac{1 - \frac{\log(2-\delta)}{\log\left(\frac{\tan^2(u^0,w_2)}{\delta}\right)}}{1+\sum_{i=1}^{\infty} \frac{1}{(i+1)!}\left(\frac{\lambda_3-\lambda_2}{g-\lambda_2}\right)^i} \frac{\log\left(\frac{\tan^2(u^0,w_2)}{\delta}\right)}{\lambda_3-\lambda_2}$$

. We note that for $\tan^2(u^0, w_2) > \delta$,

$$\frac{1 - \frac{\log(2-\delta)}{\log\left(\frac{\tan^2(u^0,w_2)}{\delta}\right)}}{1+\sum_{i=1}^{\infty} \frac{1}{(i+1)!}\left(\frac{\lambda_3-\lambda_2}{g-\lambda_2}\right)^i} < 1$$

. If $\tan^2(u^0, w_2) < \delta$, then the lower bound becomes negative. This gives us the desired result, where $C < \frac{g-\lambda_2}{2}$.

$$k > \left| \frac{C}{\lambda_3-\lambda_2} \log\left(\frac{\tan^2(u^0,w_2)}{\delta}\right) \right|_+$$

□

We see that the number of iterations required depends on the quality of the initial guess in our multilevel structure, as well as the chosen tolerance. We will analyze the behavior of k in greater detail for the case of graphs resulting from elliptic PDE discretizations in Section 3.6.

3.5.2 Alternate Refinement Strategies

We now present alternative refinement strategies to the power iteration technique detailed above. For power iteration, an initial guess $\tilde{y}_0$ is multiplied by $gI - L$ iteratively, until a given tolerance is reached at each level. We note that this is very similar to the modified Richardson iteration. We present this concept in the form of a lemma.

Lemma 3.5.3. *Let L be the Laplacian of some graph $G = (V, E)$. Let $x^{(0)}$ be a vector such that $(\mathbf{1}, x^{(0)}) = 0$. Solving the linear system $Lx = 0$ with modified Richardson iteration, $\omega = \frac{1}{g}$, stopping criterion $||x^{(i+1)} - x^{(i)}|| < tol$, and initial guess $x^{(0)}$ is equivalent to solving for the Fiedler vector of L using power iteration on the matrix $gI - L$, with stopping criterion $||x^{(i+1)} - x^{(i)}|| < tol$, and initial guess $x^{(0)}$.*

Proof. Recall that for $Ax = b$, modified Richardson iteration is given by $x^{(i+1)} = x^{(i)} + \omega(b - Ax^{(i)})$. Therefore, for $Lx = 0$ and $\omega = \frac{1}{g}$, modified Richardson gives us $x^{(i+1)} = x^{(i)} - g^{-1}(Lx^{(i)}) = (I - g^{-1}L)x^{(i)}$. However, at each step, $x^{(i+1)}$ is normalized. Therefore multiplying the right hand side by a constant does not change the end result. Multiplying by g gives us $x^{(i+1)} = (gI - L)x^{(i)}$. This is precisely power iteration for the matrix $gI - L$. □

We see that the power iteration can be shown in the form of a linear iterative method. For this reason, it is natural to consider a more general class of linear iterative methods of the form $x^{(i+1)} = x^{(i)} + B(b - Ax^{(i)})$ for $Lx = 0$. In addition to modified Richarson (power iteration), we

can also use methods such as Jacobi method and forward, backward, and symmetric Gauss-Seidel method.

We decompose the graph Laplacian L into three matrices $L = \mathcal{U} + \mathcal{D} + \mathcal{L}$, where $\mathcal{U}$ is strictly upper triagular, $\mathcal{D}$ is diagonal, and $\mathcal{L}$ is strictly lower triangular. For the sake of the reader, we give a reminder as to what B is defined as for each of these linear methods:

Jacobi Method: $B = \mathcal{D}^{-1}$
Forward Gauss-Seidel: $B = (\mathcal{U} + \mathcal{D})^{-1}$
Backward Gauss-Seidel: $B = (\mathcal{D} + \mathcal{L})^{-1}$
Symmetric Gauss-Seidel: $B = (\mathcal{D} + \mathcal{U})^{-1} + (\mathcal{D} + \mathcal{L})^{-1} - (\mathcal{D} + \mathcal{U})^{-1} L (\mathcal{D} + \mathcal{L})^{-1}$

We consider these linear iterative methods as alternatives to power iteration. However, rather than give a specific stopping criterion, we perform a fixed number of iterations on each level. The number of iterations at level i is given by 2^{i-1}, so more iterations are performed on the coarse grids, and less on the finer grid. This is similar to what occurs with the given power iteration stopping criterion. However, this is a more traditional refinement representation for a cascadic multigrid algorithm. We consider these smoothers in addition to the power iteration in our numerical tests.

We note that these linear iterative methods require stricter conditions on our Laplacian L. In particular, we require D to be invertible. Therefore we require every node to be connected to some vertex. However, this is not guaranteed by our heavy edge coarsening procedure. Therefore, we must check that each coarse Laplacian has a positive diagonal. If this is not the case, we attach the isolated node v_i to vertex v_{i+1}. This is not required for the power iteration, because the Richardson method does not depend on the invertability of the diagonal.

3.6 Convergence Analysis of CMG for Elliptic Eigenvalue Problems

In Section 3.3, we introduced the CMG method for computing the Fiedler vector of a graph Laplacian. However, we use the purely algebraic coarsening strategy (see Section 3.4) to construct the hierarchical structure; hence, similar to the AMG method for the Poisson problem, the convergence analysis for such purely algebraic CMG method is rather difficult. In order to provide a certain level of theoretical support to our proposed CMG method, in this section, we discuss the geometric CMG (GCMG) method for the elliptic eigenvalue problems. Because we are interested in the graph Laplacian, we consider the following elliptic eigenvalue problem with Neumann boundary condition,

$$\begin{cases} -\Delta u = \lambda u, & \text{on } \Omega \\ \dfrac{\partial u}{\partial n} = 0, & \text{on } \partial\Omega \end{cases} \tag{3.2}$$

where $\Omega \in \mathbb{R}^d$ is a polygonal Lipschitz domain. We only consider the two- and three- dimensional case here because of the simplicity of the chapter and its close connection with graph Laplacian. However, the GCMG method we discussed here can be naturally applied for high diminutional cases. Using the standard Sobolev space $H^1(\Omega)$, we consider the weak formulation of (3.2) as follows: find $(\lambda, w) \in \mathbb{R} \times H^1(\Omega)\backslash\mathbb{R}$, such that,

$$a(w, v) = \lambda(w, v), \quad \forall\ v \in H^1(\Omega), \tag{3.3}$$

where the bilinear form $a(u, v) = (\nabla u, \nabla v)$, and $(\cdot, \cdot)$ is the standard L^2 inner product. Here, the bounded symmetric bilinear form $a(\cdot, \cdot)$ is coercive on the quotient space $H^1(\Omega)$, and therefore, induces an energy-norm as follows,

$$\|u\|_a^2 = a(u, u), \quad \forall\ u \in H^1(\Omega)\backslash\mathbb{R}. \tag{3.4}$$

Moreover, we denote the L^2-norm by $\|\cdot\|$ as usual. Similar to the eigenvalues for the graph Laplacian, $\lambda = 0$ is also a eigenvalue of the eigenvalue problem (3.3), Again, we are interested in approximating the second smallest eigenvalue of (3.3) and its corresponding eigenfunction space.

Given a nested family of quasi-uniform triangulations $\{\Gamma_j\}_{j=0}^J$, namely,

$$\frac{1}{c}2^{j-J} \leq h_j = \max_{T\in\Gamma_j} \operatorname{diam}(T) \leq c2^{j-J},$$

the spaces of linear finite elements are

$$V_j = \{u \in C(\Omega) : u|_T \in P_1(T), \forall\ T \in \Gamma_j\},$$

where $P_1(T)$ denotes the linear functions on the triangle T. We have

$$V_J \subset V_{J-1} \subset \cdots \subset V_0 \subset H^1(\Omega).$$

The finite element approximations of (3.3) on each level are as follows: find $(\lambda_j, w_j) \in \mathbb{R} \times V_j$, such that,

$$a(w_j, v_j) = \lambda_j(w_j, v_j), \quad \forall\ v_j \in V_j. \tag{3.5}$$

Again, we are interested in approximating the second smallest eigenpair $(\lambda_0^{(2)}, w_0^{(2)})$ on the finest level. Moreover, we can define an operator A_j by $a(u_j, v_j) = (A_j u_j, v_j)$, $\forall u_j, v_j \in V_j$. In addition, we define E_j to be the orthogonal eigenvalue projection of the energy space onto the Galerkin eigenvectors on V_j (see [1] for the formal definition in terms of complex integration).

We assume the elliptic eigenvalue problem has $H^{1+\alpha}$-regularity, i.e., the eigenvalue function $w \in H^{1+\alpha}$ for some $0 < \alpha \leq 1$. Then we have the following error estimates regarding the standard finite element approximation of the elliptic eigenvalue problem, which is taken from the work of Babuska and Osborn [1].

Lemma 3.6.1. *Assume that $(\lambda_j, w_j) \in (\mathbb{R} \times V_j)$ is the finite element eigenpair of (3.5). Then we have*

(i) $|\lambda - \lambda_j| \leq C h_j^{2\alpha}$,

(ii) there exists an eigenfunction w corresponding to λ, such that

$$\|w - w_j\|_a \leq C h_j^{\alpha}, \tag{3.6}$$

where C is a constant that does not depend on the mesh size.

In addition, we can prove the following *approximation property.*

Lemma 3.6.2. *Let $(\lambda_j^{(l)}, w_j^{(l)}) \in (\mathbb{R} \times V_j)$ be a finite element eigenpair of (3.5), with $\lambda^{(l)}$ a simple eigenvalue. Then, for sufficiently small h_j, we have*

$$\|(I - E_{j+1}) w_j^{(l)}\|_{H^{1-\alpha}} \leq C h_j^{\alpha} \|(I - E_{j+1}) w_j^{(l)}\|_a. \tag{3.7}$$

Proof. We begin by proving a relation between the Ritz projection P_{j+1} and the eigenvalue projection E_{j+1}, as defined in [1]. Namely, we aim to show

$$\|(I - E_{j+1}) w_j^{(l)}\|_{L^2} \leq C \|(I - P_{j+1}) w_j^{(l)}\|_{L^2}. \tag{3.8}$$

We note that $P_{j+1} w_j^{(l)} \in V_{j+1}$, and can be represented as

$$P_{j+1} w_j^{(l)} = \sum_{i=1}^{N_{j+1}} (P_{j+1} w_j^{(l)}, w_{j+1}^{(i)})_{L^2} \; w_{j+1}^{(i)}.$$

Letting $\alpha_l = (P_{j+1} w_j^{(l)}, w_{j+1}^{(l)})_{L^2}$,

$$\|P_{j+1} w_j^{(l)} - \alpha_l w_j^{(l)}\|_{L^2}^2 = \sum_{i \neq l} (P_{j+1} w_j^{(l)}, w_{j+1}^{(i)})_{L^2}^2.$$

We have the identity

$$(\lambda_{j+1}^{(i)} - \lambda_j^{(l)})(P_{j+1} w_j^{(l)}, w_{j+1}^{(l)})_{L^2} = \lambda_j^{(l)} ((w_j^{(l)} - P_{j+1} w_j^{(l)}), w_{j+1}^{(l)})_{L^2},$$

since

$$\lambda_{j+1}^{(i)} (P_{j+1} w_j^{(l)}, w_{j+1}^{(i)})_{L^2} = a(P_{j+1} w_j^{(l)}, w_{j+1}^{(i)}) = a(w_j^{(l)}, w_{j+1}^{(i)}) = \lambda_j^{(i)} (P_{j+1} w_j^{(l)}, w_{j+1}^{(i)})_{L^2},$$

where we use the fact that the billinear form $a(\cdot, \cdot)$ is symmetic. In addition, if $\lambda_j^{(l)}$ is a simple

eigenvalue, for sufficiently small h_j we have that there is a separation constant d_l such that

$$\frac{|\lambda_j^{(l)}|}{|\lambda_j^{(l)} - \lambda_{j+1}^{(i)}|} \leq d_l \qquad \text{for all } i.$$

Let Q_{j+1} be the L^2 projection to V_{j+1}. We have

$$\begin{aligned}
\|P_{j+1}w_j^{(l)} - \alpha_l w_j^{(l)}\|_{L^2}^2 &\leq d_l^2 \sum_{i\neq l} (w_j^{(l)} - P_{j+1}w_j^{(l)}, w_{j+1}^{(i)})_{L^2}^2 = d_l^2 \sum_{i\neq l} (Q_{j+1}w_j^{(l)} - P_{j+1}w_j^{(l)}, w_{j+1}^{(i)})_{L^2}^2 \\
&\leq d_l^2 \sum (Q_{j+1}w_j^{(l)} - P_{j+1}w_j^{(l)}, w_{j+1}^{(i)})_{L^2}^2 = d_l^2 \|Q_{j+1}w_j^{(l)} - P_{j+1}w_j^{(l)}\|_{L^2}^2 \\
&= d_l^2 \|Q_{j+1}(w_j^{(l)} - P_{j+1}w_j^{(l)})\|_{L^2}^2 \leq d_l^2 \|w_j^{(l)} - P_{j+1}w_j^{(l)}\|_{L^2}^2
\end{aligned}$$

Therefore,

$$\|w_j^{(l)} - \alpha_l w_j^{(l)}\|_{L^2} \leq \|w_j^{(l)} - P_{j+1}w_j^{(l)}\|_{L^2} + \|P_{j+1}w_j^{(l)} - \alpha_l w_j^{(l)}\|_{L^2} \leq (1+d_l)\|w_j^{(l)} - P_{j+1}w_j^{(l)}\|_{L^2}$$

It can be easily verified that $|\alpha_l - 1| \leq \|w_j^{(l)} - \alpha_l w_{j+1}^{(l)}\|_{L^2}$ if we choose the unit eigenfunctions $w_j^{(l)}$ and $w_{j+1}^{(l)}$ such that $\alpha_l \geq 0$. This gives us

$$\begin{aligned}
\|(I - E_{j+1})w_j^{(l)}\|_{L^2} &\leq \|w_j^{(l)} - w_{j+1}^{(l)}\|_{L^2} \leq \|w_j^{(l)} - \alpha_l w_{j+1}^{(l)}\|_{L^2} + |\alpha_l - 1| \\
&\leq 2\|w_j^{(l)} - \alpha_l w_{j+1}^{(l)}\|_{L^2} \leq 2(1+d_l)\|w_j^{(l)} - P_{j+1}w_j^{(l)}\|_{L^2}
\end{aligned}$$

This proves (3.8). Based on this result, we have

$$\begin{aligned}
\|(I - E_{j+1})w_j^{(l)}\|_{L^2} &\leq C\|(I - P_{j+1})w_j^{(l)}\|_{L^2} \\
\text{(see Theorem 8.4.14 of [18])} \quad &\leq Ch\|(I - P_{j+1})w_j^{(l)}\|_a \\
&\leq Ch\|(I - E_{j+1})w_j^{(l)}\|_a,
\end{aligned}$$

where the last inequality follows from noting that $\|(I - P_{j+1})u_j\|_a = \inf_{v\in V_{j+1}} \|u_j - v\|_a$. By an interpolation argument, the desired result follows. □

Based on the nested spaces $V_J \subset V_{J-1} \subset \cdots \subset V_0$, the GCMG method for eigenvalue problems seeks to solve the eigenvalue problem exactly on the coarse grid V_J, and interpolate and smooth the approximation back to the fine grid V_0. In this section, we consider the GCMG method, and therefore, the geometric prolongation and restriction are used in our algorithm, and will be omitted as usual. Our cascadic Algorithm 1 can be framed as follows:

Algorithm 4: Geometric Cascadic Multgrid Method for Elliptic Eigenvalue Problem

if $j = J$ *(coarsest level)* **then**
  solve $a(w_J, v_J) = \lambda_{w_J, v_J}$ exactly, and let $u_J := w_J^{(2)}$.
else
  $u_j = (I - \omega_j A_j)^{k_j} u_{j+1}$, where $\omega_j = \|A_j\|_\infty^{-1}$. (with appropriate scaling)
  $\lambda_j = \frac{a(u_j, u_j)}{(u_j, u_j)}$.

We consider the uniform convergence of the proposed GCMG method. Our analysis will follow the standard convergence analysis for the CMG method for elliptic partial differential equations. We begin by recalling the following lemma.

Lemma 3.6.3. *For any $k \in \mathbb{Z}_+$, we have* $\max_{t\in[0,1]} t(1-t)^k < \frac{1}{k+1}$.

This is a common result from Calculus, and is used often in multigrid literature. Denoting by $S_j = I - \omega_j A_j$ the error prolongation of the Richardson smoother, we have the following *smoothing property.*

Lemma 3.6.4. *Let k_j be the number of smoothing steps on level j, we have*

$$\|S_j^{k_j} v_j\|_a \leq C \frac{h_j^{-\alpha}}{k_j^{\alpha/2}} \|v_j\|_{H^{1-\alpha}}, \quad \forall\ v_j \in V_j \backslash \mathbb{R}. \tag{3.9}$$

Proof. Consider $S = I - \omega A$, with $\omega = ||A||_\infty^{-1}$. We have that A is Hermitian and positive semi-definite, by the properties of a graph Laplacian. For u such that $(u, \mathbf{1}) = 0$, we have that A is positive definite in the subspace $\{u | (u, \mathbf{1}) = 0\}$. We have

$$\begin{aligned} ||S^\nu u||_a^2 &= \big((I - \omega A)^\nu u, (I - \omega A)^\nu u\big)_a \\ &= \big(h^{-2} A (I - \omega A)^\nu u, (I - \omega A)^\nu u\big) \\ &= \omega^{-1} \big(\omega A (I - \omega A)^{2\nu} u, u\big) \end{aligned}$$

Noting that $\omega ||A|| \leq 1$, $\omega^{-1} \eqsim h^{-2}$ and making use of Lemma 3.6.3, we obtain

$$||S^\nu u||_a^2 \lesssim h^{-2} \eta_0(2\nu) ||u||^2$$

. This gives us

$$\|S_j^{k_j} v_j\|_a \leq C \frac{h_j^{-1}}{k_j^{1/2}} \|v_j\|, \quad \forall\ v_j \in V_j \backslash \mathbb{R}$$

. Recalling that we have $\|S_j^{k_j} v_j\|_a \leq C \|v_j\|_a, \quad \forall\ v_j \in V_j$, the desired result follows. □

We are now able to show the uniform convergence of our GCMG Algorithm 4 under suitable conditions.

Lemma 3.6.5. *For sufficiently small h_J, the error of the GCMG Algorithm 4 with the Richardson smoother for the eigenvector can be estimated by*

$$\|w_0^{(2)} - u_0\|_a \leq C \sum_{j=0}^{J-1} \frac{h_j^\alpha}{k_j^{\alpha/2}},$$

and for the eigenvalue, by

$$|\lambda_0 - \lambda_0^{(2)}| \leq C \Big(\sum_{j=0}^{J-1} \frac{h_j^\alpha}{k_j^{\alpha/2}} \Big)^2 \tag{3.10}$$

Proof. Let us first assume we have some approximate eigenfunction u_{j+1} on V_{j+1}. We can represent this approximation as follows:

$$u_{j+1} = w_{j+1}^{(2)} + e_{j+1}$$

where $w_{j+1}^{(2)}$ is an eigenfunction in the eigenspace corresponding to the eigenvalue $\lambda_{j+1}^{(2)}$, and e_{j+1} is the error. Because $w_{j+1}^{(2)}$ is an approximation of $w^{(2)}$ on level $j+1$, there exists an approximate eigenfunction $\bar{w}_j^{(2)}$ on level j, such that $w_{j+1}^{(2)} := E_{j+1}\bar{w}_j^{(2)}$ and $\|\bar{w}_j^{(2)} - w_{j+1}^{(2)}\|_a \leq C h_{j+1}^\alpha$. Then we have

$$u_{j+1} = \bar{w}_j^{(2)} + (w_{j+1}^{(2)} - \bar{w}_j^{(2)}) + e_{j+1}.$$

Because $u_j = S_j^{k_j} u_{j+1}$, we have

$$\begin{aligned} u_j &= S_j^{k_j}\bar{w}_j^{(2)} + S_j^{k_j}(w_{j+1}^{(2)} - \bar{w}_j^{(2)}) + S_j^{k_j} e_{j+1} \\ &= (\frac{\omega_j^{-1} - \lambda_j^{(2)}}{\omega_j^{-1}})^{k_j}\bar{w}_j^{(2)} + S_j^{k_j}(w_{j+1}^{(2)} - \bar{w}_j^{(2)}) + S_j^{k_j} e_{j+1} \\ &:= w_j^{(2)} + e_j, \end{aligned}$$

where $w_j^{(2)} := (\frac{\omega_j^{-1} - \lambda_j^{(2)}}{\omega_j^{-1}})^{k_j}\bar{w}_j^{(2)}$ and $e_j := S_j^{k_j}(w_{j+1}^{(2)} - \bar{w}_j^{(2)}) + S_j^{k_j} e_{j+1}$. Therefore, noting that $(w_{j+1}^{(2)} - \bar{w}_j^{(2)}, \mathbf{1}) = 0$, we have

$$\begin{aligned} \|e_j\|_a &\leq \|S_j^{k_j}(\bar{w}_j^{(2)} - w_{j+1}^{(2)})\|_a + \|S_j^{k_j} e_{j+1}\|_a \\ \text{(Lemma 3.6.4)} &\leq C \frac{h_j^{-\alpha}}{k_j^{\alpha/2}} \|\bar{w}_j^{(2)} - w_{j+1}^{(2)}\|_{H^{1-\alpha}} + \|e_{j+1}\|_a \\ \text{(Approximation property (3.7))} &\leq C \frac{1}{k_j^{\alpha/2}} \|\bar{w}_j^{(2)} - w_{j+1}^{(2)}\|_a + \|e_{j+1}\|_a \end{aligned}$$

Summing from $j = J-1$ to 0, and note that $e_J = 0$, we have

$$\|w_0^{(2)} - u_0\|_a = \|e_0\|_a \leq C \sum_{j=0}^{J-1} \frac{1}{k_j^{\alpha/2}} \|\bar{w}_j^{(2)} - w_{j+1}^{(2)}\|_a \leq C \sum_{j=0}^{J-1} \frac{h_j^\alpha}{k_j^{\alpha/2}}.$$

Moreover, using the identity

$$\lambda_0 - \lambda_0^{(2)} = \frac{a(w_0^{(2)} - u_0, w_0^{(2)} - u_0)}{(u_0, u_0)} - \lambda_0^{(2)} \frac{(w_0^{(2)} - u_0, w_0^{(2)} - u_0)}{(u_0, u_0)},$$

we have

$$|\lambda_0 - \lambda_0^{(2)}| \leq C(\sum_{j=0}^{J-1} \frac{h_j^\alpha}{k_j^{\alpha/2}})^2 \tag{3.11}$$

□

Because $2^j h_0/C \leq h_j \leq C2^j h_0$, we consider $k_j = \beta^j k_0$ for some fixed $\beta > 0$. We have the following.

Theorem 3.6.6. *Let the number of smoothing steps on level j be given by $k_j = \beta^j k_0$. If h_J is sufficiently small, then the error of the GCMG method for the eigenvector can be estimated by*

$$\|w_0^{(2)} - u_0\|_a \leq \begin{cases} C\frac{1}{1-(4/\beta)^{\alpha/2}} \frac{h_0^\alpha}{k_0^{\alpha/2}}, & \text{if } \beta > 4, \\ CJ\frac{h_0^\alpha}{k_0^{\alpha/2}}, & \text{if } \beta = 4. \end{cases}$$

and for the eigenvalue, by

$$|\lambda_0^{(2)} - \lambda_0| \leq \begin{cases} C(\frac{1}{1-(4/\beta)^{\alpha/2}})^2 \frac{h_0^{2\alpha}}{k_0^\alpha}, & \text{if } \beta > 4, \\ CJ^2 \frac{h_0^{2\alpha}}{k_0^\alpha}, & \text{if } \beta = 4. \end{cases}$$

Proof. The estimates follows directly from the following estimation

$$\sum_{j=0}^{J-1} \frac{h_j^\alpha}{k_j^{\alpha/2}} \leq C \frac{h_0^\alpha}{k_0^{\alpha/2}} \sum_{j=0}^{J-1} (\frac{4}{\beta})^{\frac{j\alpha}{2}}$$

□

What remains to be considered is the computational complexity. Assuming still that $k_j = \beta^j k_0$ for some fixed $\beta > 0$, we have the following corollary.

Corollary 3.6.7. *Let the number of smoothing steps on level j be given by $k_j = \beta^j k_0$, then the computational cost of the GCMG method is proportional to*

$$\sum_{j=1}^{J} k_j n_j \leq \begin{cases} C\frac{1}{1-\beta/2^d} k_0 n_0, & \text{if } \beta < 2^d, \\ CJk_0 n_0, & \text{if } \beta = 2^d. \end{cases}$$

Proof. The result follows naturally from noting that $2^{dj}/c \leq n_j \leq c2^{dj}$ and observing that

$$\sum_{j=1}^{J} k_j n_j \leq ck_0 n_0 \sum_{j=0}^{J-1} (\frac{\beta}{2^d})^j$$

□

We see that if we set β to be $4 < \beta < 2^d$, our results regarding accuracy and complexity do not contradict. Therefore, we see that for $d = 3$ our algorithm is optimal. However, for $d = 2$, either the accuracy or complexity must deteriorate logarithmically.

3.7 Numerical Results

We now perform numerical tests. We investigate values of k_{HEC} in practice and compare the speed of our eigensolver to the standard Lanczos algorithm. In addition, we take our eigensolver and use it to solve the graph partitioning problem. We use our eigensolver in the spectral bisection method. We perform numerical tests on a variety of different graphs (listed in Table 3.2).

Table 3.2. Details of test graphs used

Graph	Graph Type	Number of Vertices	Number of Edges
144	3D Finite Element Mesh	144649	1074393
598a	3D Finite Element Mesh	110971	741934
auto	3D Finite Element Mesh	448695	3314611
bcsstk30	3D Stiffness Matrix	28294	1007284
bcsstk32	3D Stiffness Matrix	44609	985046
brack2	3D Finite Element Mesh	62631	366559
m14b	3D Finite Element Mesh	214765	3358036
rotor	3D Finite Element Mesh	99617	662431
troll	3D Stiffness Matrix	213453	5885829
wave	3D Finite Element Mesh	156317	1059331

We compare our eigensolver with heavy edge coarsening to the random matching coarsening technique. In addition, we also investigate the different refinement processes presented, with heavy edge coarsening as the coarsening scheme. In particular, we consider power iteration, Jacobi method, and Symmetric Gauss-Seidel. Finally, we look into the computational run time for our cascadic algorithm as a partitioner for a number of different matrix sizes. All of our computations were performed on a HP Pavilion dv5 Notebook PC with a 2.40 GHz AMD Turion II P540 Dual-Core Processor. All of the algorithms tested in this section were implemented and tested in MATLAB. The .m files used in this chapter can be found online at http://www.personal.psu.edu/jcu5018, under the publications tab.

We begin by looking at the value of k_{HEC} for one iteration of HEC for each graph. In addition, we consider the number of iterations needed to coarsen to less than 25 vertices. These results are given in Table 3.3.

On average, our intial values of k_{HEC} appear to be roughly in the 0.1-0.2 range. In terms of speed, this is significantly quicker than any matching technique. Only about 6-8 iterations are required. What remains to be shown is how well the coarse graphs approximate our fine graph. This will be made apparent in the speed of the eigensolver.

Table 3.3. Test values of k_{HEC}

Graph	Number of Subgraphs	k^0_{HEC} Value
144	7	0.1893
598a	7	0.2024
auto	8	0.1742
bcsstk30	5	0.0700
bcsstk32	6	0.1386
brack2	7	0.2066
m14b	8	0.1969
rotor	7	0.1819
troll	7	0.1172
wave	7	0.1736

We consider the performance of our eigensolver against the Lanczos algorithm. In Table 3.4 we report the run times in seconds for each graph, along with a measure of the error in the approximate eigenvector, given by $\|(L - \tilde{r}I)\tilde{y}\|_\infty$, where $\tilde{y}$ is the approximate eigenvector, and $\tilde{r}$ is the corresponding approximate eigenvalue.

Table 3.4. Test results for eigensolver algorithm

	Lanczos Algorithm		Cascadic Eigensolver	
Graph	Run Time	Error	Run Time	Error
144	189.50	0.0092	2.17	0.0019
598a	134.33	0.0099	1.53	0.0042
auto	***	***	7.05	0.0030
bcsstk30	117.83	0.0098	0.46	0.0759
bcsstk32	***	***	0.69	0.0923
brack2	173.20	0.0090	0.74	0.0037
m14b	***	***	3.04	0.0053
rotor	***	***	1.14	0.0223
troll	***	***	3.24	0.0317
wave	***	***	1.98	0.0035

Note that our eigensolver significantly outperforms the Lanczos algorithm. Boxes with triple asteriks (***) denote computations which failed to execute in 300 seconds. Usually these computations fail, as a result of insufficient memory.

We now consider using our cascadic eigensolver in recursive bisection. We perform a 64-way partition of each graph. We compare the effect of our coarsening scheme to random matching. We use the same power iteration refinement and bisection procedure. In addition, we compare power iteration, Jacobi method, and Symmetric Gauss-Seidel as refinement procedures, all with heavy edge coarsening as the coarsening technique. We compare both the edge cut and run time of each method. The results are given in Table 3.5 and 3.6.

Table 3.5. Test results for coarsening algorithms

	Heavy Edge Coarsening		Random Matching	
Graph	Run Time	Edge Cut	Run Time	Edge Cut
144	11.85	101244	116.5	167012
598a	9.1	72608	84.08	115728
auto	38.93	223866	554.2	569894
bcsstk30	4.22	219101	53.82	252915
bcsstk32	4.69	138640	77.7	184661
brack2	4.8	36903	36.15	61736
m14b	18.2	130568	205.9	310378
rotor	7.96	70869	76.67	174038
troll	23.07	576780	543.85	1308683
wave	12.4	108756	125.65	198843

Looking at the coarsening results, we observe that with heavy edge coarsening the run times are significantly faster. In addition, the edge cuts are better as well. This shows that heavy edge coarsening is significantly better than the random matching procedure.

Table 3.6. Test results for refinement algorithms

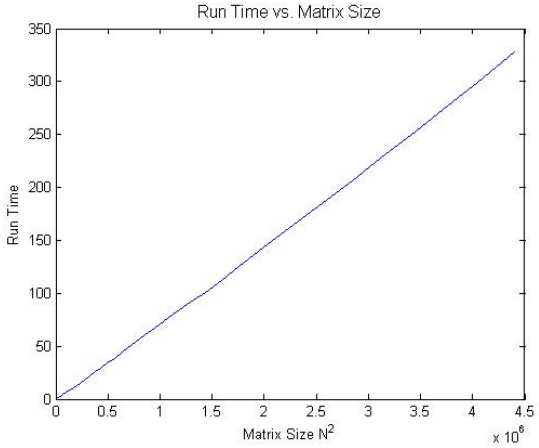
	Power Iteration		Jacobi Method		Symmtric Gauss-Seidel	
Graph	Run Time	Edge Cut	Run Time	Edge Cut	Run Time	Edge Cut
144	11.85	101244	10.48	101721	10.98	98108
598a	9.1	72608	8.02	73736	8.28	69632
auto	38.93	223866	34.79	217197	36.88	211246
bcsstk30	4.22	219101	2.72	214679	2.85	216713
bcsstk32	4.69	138640	3.69	135969	3.79	130842
brack2	4.8	36903	4.25	35821	4.41	34338
m14b	18.2	130568	16.56	129331	16.84	124367
rotor	7.96	70869	6.92	68561	6.99	64160
troll	23.07	576780	18.91	533655	20.01	522397
wave	12.4	108756	10.83	110530	11.07	110822

When we look at the refinement procedures, we note that the Jacobi refinement is the fastest and Symmetric Gauss-Seidel is very close, while also having the lowest edge cuts of the group. We note that the edge cut of a graph for a spectral method is an indirect measure of the error of the underlying eigensolver. Generally, a consistently lower edge cut implies that the underlying eigensolver is more accurate.

We now consider using our cascadic algorithm as a partitioning algorithm for the standard two-dimensional Laplacian problem with Neumann boundary conditions. We look at a 64 way partition of the structured graph for a variety of different grid sizes N^2 varying from size $O(10^2)$ up to size $O(10^6)$. We used a sample size of 50, with the inverse of our step size $h^{-1} = N$ given by $N = \lfloor 10^{ln(\frac{i}{2}+e)} \rfloor$, where $i = 1, 2, ..., 50$. We report the run time for these different sizes. The graph,

with respect to matrix size, is given below.

Figure 3.2. Plot of Run Time vs. Matrix Size for the 2D Laplacian Problem



We see that the plot of run time vs. matrix size (Figure 3.2) appears to be almost perfectly linear. Upon examination of the sample, we report a correlation of $R = 0.9998$. This provides strong numerical evidence that, as a partitioning algorithm, our cascadic eigensolver is linear with respect to time.

3.8 Conclusion

In this chapter, we have presented a fast algorithm for approximately computing the Fiedler vector of a graph Laplacian. We introduced a new coarsening procedure, called heavy edge coarsening. We note the speed with which the procedure coarsens, and the quality of coarse level graphs. The main contribution to the speed of the algorithm was a result of the implementation of the heavy edge coarsening procedure.

In addition to being a fast coarsening procedure, the heavy edge coarsening algorithm is also significantly easier to implement than other techniques of a similar type, such as heavy edge matching and its variants (HEM and HEM*) [26, 27]. In terms of simplicity, it is along the lines of random matching. Looking at refinement procedures, we considered a more standard eigensolver in the form of power iteration, as well as linear iterative methods. As an eigensolver, the combination of heavy edge coarsening and power iteration in a cascadic multigrid method proves to be a fast algorithm for finding the Fiedler vector of graph Laplacians.

Chapter 4

A Space-Time Method for the Numerical Pricing of Barrier Options

4.1 Introduction

The computation of the fair value price of a given barrier option is a relevant problem in finance. Numerical techniques are particularly relevant, mainly due to the lack of closed-form representations for their valuation under the majority of models. In addition, there exist nuances for the case of the barrier option that separates it from the general vanilla options.

There are a number of different models that can be used to determine the fair value of a barrier option. The most well known formulation by far is the Black-Scholes (BS) model. While it is fairly simplistic (in comparison to its counterparts), and exhibits a closed form solution for barrier options, it does not take into account the volatility smile observed in practice. In fact, the model is especially inaccurate for the case of barrier options. The difference between the assumptions of the Black-Scholes model and what occurs in practice produces errors that are particularly relevant for barrier options. In addition, barrier options are extremely sensitive to mispecifications in parameter estimation, resulting in large pricing errors. This is a result of discontinuitites in the payoffs of many barrier options, resulting in large Gamma ($\Gamma = \frac{\partial^2 V}{\partial S^2}$), and, therefore, large Vega ($\nu = \frac{\partial V}{\partial \sigma}$) [24].

In cases where a deterministic or surface volatility are used, exact solutions no longer exist. However, due to reasons stated above, such volatility models are particularly useful for barrier options. In addition to the Black-Scholes model, we give treatments of the constant elasticity of variance (CEV) model and Heston's model as well.

There exists a number of different techniques for pricing options under such models. In particular, the two main techniques used in practice are to estimate the price by Monte Carlo methods, or to estimate the price by solving the model's corresponding partial differential equation (PDE). In this chapter, we consider the latter case exclusively. For a treatment of Monte Carlo methods for financial engineering, refer to [16].

The partial differential equation approach requires the computation of the solution of a second order linear parabolic PDE. For this type of PDE, there are a number of different possible approaches. For our purposes, we consider the method of lines, with a finite differences discretization in space. In practice, this technique is extremely popular, especially in financial circles. However, for completeness, we introduce the finite element framework as well.

We consider applying multigrid (MG) methods to the discretization. We treat the entire grid simultaneously and introduce an adaptive restriction procedure that depends on the problem's characteristics and discretized grid size. At each level, the restiction operator is determined through local Fourier analysis (sometimes referred to as local mode analysis). Depending on the grid size, we choose to coarsen either in space or time. Simultaneous coarsening fails for a large class of grids, as the anisotropy ratio approaches zero on coarser grids.

Treating space and time simultaneously allows for full parallelization of the algorithm. For a grid with N points in space and M in time, multigrid in space with time-stepping has parallel complexity $O(M \log N)$, whereas treating the grid as a whole, with a suitable smoother, results in parallel complexity $O(\log M + \log N)$. For fine grids, this difference is extremely pronounced; even for extremely large problems the parallel complexity of our algorithm is nominal, allowing for fast computation of even the largest problems.

We begin by introducing the mathematical framework of our problem. We give the numerical details of our method for the Black-Scholes operator by finite differences. We introduce the finite element method as well. We detail the technique used to determine the restriction operator and give numerical results. We show our method to be robust to other models and consider the CEV and Heston's model. In addition, we give a discussion of our method in general, with respect to its advantages and disadvantages.

4.2 Mathematical Framework

We consider a specific type of derivative, in the form of barrier options. The intuition behind barrier options is to take a given derivative and put an additional condition on its movement. In particular, we set a stock level B and observe the stock's movement over the life of the option $[0, T]$. Depending on the type of barrier option, the option will become worthless if the extra condition involving the stock price level B is not satisfied. Thus a barrier option will have the payoff of a call or a put, unless it is rendered worthless by an additional condition. These conditions depend on the type of barrier option considered.

There are predominantly eight different types of barrier options. The different types of barrier options depends on three different choices. The first choice is whether the option is a call or a put. The remaining two are as to whether the option is an up-and-out, up-and-in, down-and-out, or down-and-in barrier option. We will assume our barrier options to contain only one barrier (eliminating more complicated barriers, such as a double knock-out option, from the conversation). In addition, we consider only continuous barriers and assume the option to be European in nature. We have the following definition.

Definition 4.2.1. *A* **barrier option** *is a call or put derivative with underlying S, with an additional condition for the path of the underlying S_t with respect to a given level of stock price B. An up-and-out barrier option is worthless if there exists a $t \in [0, T)$ such that $S_t \geq B$. An up-and-in barrier option is worthless if there does not exist a $t \in [0, T)$ such that $S_t \geq B$. A down-and-out barrier option is worthless if there exists a $t \in [0, T)$ such that $S_t \leq B$. A down-and-in barrier option is worthless if there does not exist a $t \in [0, T)$ such that $S_t \leq B$.*

We help illustrate this definition with a set of figures. We consider an up-and-out European call option. Looking at Figures 4.1 and 4.2, we see two realizations of paths of the underlying stock. In Figure 4.1, we see that the stock crosses the boundary (there exists a $t < T$ such that $S_t \geq B$). Therefore, in this case, the option expires worthless. However, in Figure 4.2 the stock does not cross the boundary. Therefore the option expires as a standard call, with value $|S_T - K|_+$.

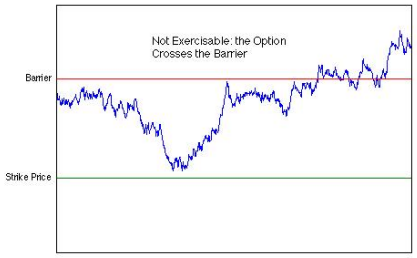


Figure 4.1. Plot of a stock that crosses the barrier

Figure 4.2. Plot of a stock that doesn't cross the barrier

We give a list of the types of barrier options in Table 4.1, along with their worth, based on interaction with the barrier. This table is similar to one found in [10]. We note that, of these eight types, there are only four independent cases. The sum of an up-and-in (or down-and-in) option and an up-and-out (or down-and-out) gives a vanilla option.

Table 4.1. Types of Single Barrier Options

Option	Type	Barrier Location	Crossed	Not Crossed
Call	Down-and-Out	Below Spot	Worthless	Standard Call
	Down-and-In	Below Spot	Standard Call	Worthless
	Up-and-Out	Above Spot	Worthless	Standard Call
	Up-and-In	Above Spot	Standard Call	Worthless
Put	Down-and-Out	Below Spot	Worthless	Standard Call
	Down-and-In	Below Spot	Standard Call	Worthless
	Up-and-Out	Above Spot	Worthless	Standard Call
	Up-and-In	Above Spot	Standard Call	Worthless

For the purposes of this chapter, we will consider up-and-out European call options exclusively.

We take our underlying to be given by

$$dS(t) = \beta(S(t))\, dt + \gamma(S(t))\, d\tilde{W}(t) \tag{4.1}$$

where $\tilde{W}(t)$, $0 \leq t \leq T$, is a Brownian motion on the risk neutral measure $\tilde{\mathcal{P}}$. Mathematically, by the Feymann-Kac formula, the fair price of an up-and-out call option at time t, $v(x,t) = \mathbb{E}^{x,t}[e^{-r(T-t)}v(x,T)]$, is the solution to the following partial differential equation:

$$\begin{cases} v_t + \mathcal{A}v - rv = 0 \quad on \quad [0,T) \times (0,B) \\ v(t,0) = v(t,B) = 0 \quad for \quad t \in [0,T) \\ v(T,x) = |x-K|_+ \quad for \quad x \in [0,B], \end{cases} \tag{4.2}$$

where $\mathcal{A}$ is given by

$$\mathcal{A} = \beta(x,t)\, \frac{\partial}{\partial x} + \frac{1}{2}\, \gamma(x,t)^2\, \frac{\partial^2}{\partial x^2}. \tag{4.3}$$

Depending on the model, the choice of $\mathcal{A}$ varies. However, for now we can assume $\beta(x,t) = rx$ and $\gamma(x,t) = \sigma x$, where r and σ are constant. This gives us the Black-Scholes partial differential equation. We save the treatment of the CEV model and Heston's model to Section 4.6.

The author notes that a significant amount of subtlety regarding the applicability of the Markov property and the Feymann-Kac Theorem has been neglected. However, it is not directly relevant to the topic of the chapter, and has been omitted. For a more detailed analysis of why the partial differential equation for the barrier option is the same as for the case of vanilla options, and the process to come to this result, refer to [39, 40]. For a more general introduction to the theory of stochastic differential equations, refer to [33].

In addition to the standard representation of the Black-Scholes PDE, through a coordinate transformation of the domain, one can obtain the standard diffusion equation (see Chapter 4 of [42]). We consider the transformation $x = e^y$, $t = T - \frac{2\tau}{\sigma^2}$. This gives us the following transformed differential equation for $v(y,\tau) = v(\ln(x), \frac{\sigma^2}{2}(T-t))$:

$$v_\tau = v_{yy} + (\frac{2r}{\sigma^2} - 1)v_x - \frac{2r}{\sigma^2}v \qquad for \quad -\infty < x < \ln(B), \quad 0 \leq \tau < \frac{\sigma^2}{2}T \tag{4.4}$$

Setting $v(y,\tau) = e^{\alpha y + \beta\tau}u(y,\tau)$, for

$$\alpha = \frac{\sigma^2 - 2r}{2\sigma^2} \tag{4.5}$$

$$\beta = \big(\frac{\sigma^2 + 2r}{2\sigma^2}\big)^2, \tag{4.6}$$

we obtain the heat equation on a transformed domain, with similar boundary conditions.

$$\begin{cases} u_\tau = u_{yy} & on \quad (-\infty, \ln(B)) \times [0, \frac{\sigma^2}{2}T) \\ u(\tau, \ln(B)) = 0 & for \quad 0 \le \tau < \frac{\sigma^2}{2}T \\ u(0, y) = e^{-\alpha y}|e^y - K|_+ & for \quad -\infty < y < \ln(B) \end{cases} \tag{4.7}$$

We will treat this formulation of the Black-Scholes PDE. We note that for any solution obtained by the representation (4.7), the corresponding fair option value can be easily obtained by multiplying by $e^{\alpha y + \beta \tau}$ and transforming $(y, \tau) \to (x, t)$. It will be shown that this formulation is much more suitable for local mode analysis, and, therefore, significantly more suitable for an adaptive space-time multigrid method.

4.3 Numerical Formulation

We begin by considering the problem of discretizing our PDE. This can be done by either finite differences or finite element. In practice, finite differences is the preferred discretization method in finance, and, for this reason, we will follow this trend for the moment. We leave the theoretical treatment of finite element to Subsection 4.3.1.

We begin by recalling the centered first and second derivative discretizations.

Lemma 4.3.1. *Let $f(x)$ be a C^2 function. Then we have:*

(i) $f'(x) = \frac{f(x+h)-f(x-h)}{2h} + O(h^2)$

(ii) $f''(x) = \frac{f(x+h)-2f(x)+f(x-h)}{h^2} + O(h^3)$

From here we can consider the discretization of (4.7) in space. Because of the infinite lower boundary in space, we must set an artificial lower boundary for our problem. We choose $-\log(B)$, which transforms to a value of $\frac{1}{B}$ in the standard domain, and, for reasonably large B, is sufficiently close to zero. The error of this approximation can be explicitly calculated by computing the difference in solution between an up-and-out and double knock-out call with lower barrier at $1/B$.

We define a partition in space $\Pi_x = \{x_1, x_2, ..., x_{N-1}, x_N\}$, where $-\log(B) = x_1 < x_2 < ... < x_{N-1} < x_N = log(B)$, $h_x = x_{j+1} - x_j$. Using Lemma 4.3.1, we obtain the following discrete formulation.

$$\begin{cases} v_t = \tilde{v}_{xx}(\tau, x_j) \quad for \quad t \in (0, T], \ 0 < j < N \\ v(t, x_0) = v(t, x_N) = 0 \qquad for \quad t \in (0, T] \\ v(0, x_j) = e^{-\alpha x_j}|e^{x_j} - K|_+ \qquad for\ all \quad 0 < j < N \end{cases} \tag{4.8}$$

where

$$\tilde{v}_{xx}(t, x_j) := \frac{v(t, x_{j+1}) - 2v(t, x_j) + v(t, x_{j-1})}{h_x^2} \tag{4.9}$$

We have now approximated our problem by a system of $N-2$ forward ordinary differential equations, with initial conditions at $t=0$. We recall the common first- and second-order numerical techniques for solving ordinary differential equation initial value problems in Table 4.2. We denote by $\Psi^{t+\tau,t}x$ the approximate solution obtained from a step of size tau.

Table 4.2. Common Numerical Techniques for Ordinary Differential Equations

Method	Formulae	Order
Explicit Euler	$\Psi^{t+\tau,t}x = x + \tau f(t,x)$	$O(h)$
Implicit Euler	$\Psi^{t+\tau,t}x = x + \tau f(t+\tau, \Psi^{t+\tau,t}x)$	$O(h)$
Midpoint	$\Psi^{t+\tau,t}x = x + \tau f(t+\frac{\tau}{2}, x+\frac{\tau}{2}f(t,x))$	$O(h^2)$
Trapezoidal Rule	$\Psi^{t+\tau,t}x = x + \frac{\tau}{2}(f(t,x) + f(t+\tau, \Psi^{t+\tau,t}x))$	$O(h^2)$

Both the explicit Euler and midpoint method are explicit in nature, with limited stability domains, and, therefore, are rarely used in practice. We will treat both the implicit Euler method and the trapezoidal rule. When applied to partial differential equations via the method of lines, the trapezoidal rule is commonly referred to as the Crank-Nicolson method. Both methods have their advantages. The Crank-Nicolson method is of higher order and is more commonly used in practice, however, it can result in oscillatory behavior in the solutions, resulting from the non-smooth (and possiblity jump) intial conditions that occur in option pricing [48]. There are a number of ways to avoid this issue, and reasonable augmentations of the method can be found in [34]. To apply these methods, we introduce a partition in time $\Pi_t = \{t_1, t_2, ..., t_{M-1}, t_M\}$, where $0 = t_1 < t_2 < ... < t_{M-1} < t_M = T$, $h_t = t_i - t_{i-1}$.

For the implicit Euler method, $v(t_i, x_j) := \Psi_{IE}^{t_i,t_{i-1}} v(t_{i-1}, x_j)$ takes the form

$$v(t_i,x_j) = v(t_{i-1},x_j) + h_t\left(\frac{v(t_i,x_{j+1}) - 2v(t_i,x_j) + v(t_i,x_{j-1})}{h_x^2}\right), \tag{4.10}$$

resulting in the stencil

$$\begin{bmatrix} 0 & 0 & 0 \\ -h_t/h_x^2 & 1+2h_t/h_x^2 & -h_t/h_x^2 \\ 0 & -1 & 0 \end{bmatrix} \tag{4.11}$$

for each grid point (t_i, x_j). Similarly, for the Crank-Nicolson method, $v(t_i, x_j) := \Psi_{CN}^{t_i,t_{i-1}} v(t_{i-1}, x_j)$ is given by

$$\begin{aligned} v(t_i,x_j) = v(t_{i-1},x_j) + \frac{h_t}{2}\Bigg(&\frac{v(t_{i-1},x_{j+1}) - 2v(t_{i-1},x_j) + v(t_{i-1},x_{j-1})}{h_x^2} \\ &+ \frac{v(t_i,x_{j+1}) - 2v(t_i,x_j) + v(t_i,x_{j-1})}{h_x^2}\Bigg), \end{aligned} \tag{4.12}$$

producing the stencil

$$\begin{bmatrix} 0 & 0 & 0 \\ -h_t/2h_x^2 & 1+h_t/h_x^2 & -h_t/2h_x^2 \\ -h_t/2h_x^2 & -1+h_t/h_x^2 & -h_t/2h_x^2 \end{bmatrix}. \tag{4.13}$$

This gives us a system of $(M-1)\times(N-2)$ equations with $(M-1)\times(N-2)$ unknowns, one for each point $v(\tau_i, x_j)$, $0 < i \leq M$, $0 < j < N$. There are a number of ways to solve such a system; we consider multigrid methods.

Multigrid (MG) methods are a general technique used to solve discrete formulations of differential equations by producing and utilizing a multilevel grid structure. Most pointwise relaxation schemes (such as Gauss-Seidel, Jacobi, etc.) have different rates of convergence for the low and high frequency components of the error. In particular, the high frequency components are tough to smooth and, for such wavelengths, the smoother may even be divergent. This implies that the different scales should be treated separately. Due to the aliasing effect of restriction operators on the frequency domain of the error, coarse grids can be used to effectively smooth the high frequency components. Interpolation between grids is used to create a multilevel structure to maintain the smoothing effects of the pointwise smoother on the low frequency components, while also smoothing the high frequency components through the coarse grids. For the reader unfamiliar with multigrid methods, we suggest [41, 45] as references. For a more advanced and rigorous treatment of multigrid, refer to [19].

There are a number of ways to perform an effective multilevel scheme. In this chapter, we consider the multigrid V-cycle. This choice is mainly for simplicity, and a W-cycle or full multigrid (FMG) cycle could also suffice, although we note that V-cycles have been shown to be better suited to parallelization. For the reader unfamiliar with such a technique, we detail the generic multigrid cycle in Algorithm 5. Taking $\gamma = 1$ gives us the V-cycle.

For $A_J u_J = f_J$, we initialize the V-cycle with $u_J = MGCYCLE(J, \gamma = 1, A_J, u_J^0, f_J, \nu_1, \nu_2)$ for some initial guess u_J^0. We can perform as many V-cycles as necessary to achieve the desired convergence.

We consider the use of Gauss-Seidel red-black smoothing, although another method (such as Jacobi, Richardson, etc.) can also been used. However, in general, Gauss-Seidel has been shown to be the most effective pointwise smoother for multigrid techniques [42]. We choose red-black ordering, rather than lexicographic, because of the ease which which it can be parallelized.

For interpolation, we implement an adaptive restriction operator, depending on the properties of the given grid. In particular, we consider the anisotropy ratio $\lambda = h_t/h_x^2$. We determine the ideal value of h_t/h_x^2, denoted by λ_a, for a given problem through local Fourier analysis, and use it to determine whether coarsening in space or time should be performed on a given grid. The stencils for our space, time, and simultaneous space-time restriction operators, respectively, are given as follows.

$$\frac{1}{4}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \frac{1}{8}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix} \tag{4.14}$$

Algorithm 5: Multigrid Cycle $u_k^{m+1} = MGCYCLE(k, \gamma, A_k, u_k^m, f_k, \nu_1, \nu_2, \gamma)$

Pre-Smoothing:
$\tilde{u}_k^m = S^{\nu_1} u_k^m$
Coarse-Grid Correction:
$r_k = f_k - A_k \tilde{u}_k^m$
$r_{k-1} = I_k^{k-1} r_k$
Approximate Coarse Solution:
if $k = 1$ **then**
Solve $A_0 e_0 = r_0$ Exactly
$e_0 = A_0^{-1} r_0$
else
Solve $A_{k-1} e_{k-1} = r_{k-1}$ Approximately with γ MG Cycle Calls
$e_{k-1} = MGCYCLE^{\gamma}(k-1, \gamma, 0, A_{k-1}, r_{k-1}, \nu_1, \nu_2)$
Interpolate Correction:
$e_k = I_{k-1}^k e_{k-1}$
$u_k^{CGC} = \tilde{u}_k + e_k$
Post-Smoothing:
$u_k^{m+1} = S^{\nu_2} u_k^{CGC}$

Note that for the time stencil, our restriction is asymmetric, so as to not transfer any information backward in time. The prolongation operators are taken to be the adjoints of the restriction operators. The process used to determine the value of λ_a in practice is the subject of Section 4.4.

4.3.1 Finite Element Framework

As mentioned earlier, rather than implement finite differences, a finite element technique could be used in the discretization. We introduce the finite element framework. We begin by noting the differences in finite differences and finite element in practice. In general, finite element tends to be preferred for irregular domains. However, for rectangular domains, finite differences is easier to implement. Finite element is considered to be the more mathematically rigorous technique, with a much richer theory to support it. In addition, a finite element discretization tends to be a better approximation than its finite differences counterpart.

We consider the variational formulation of (4.7). Denote by $a^{BSM}(\cdot,\cdot) : H^1(\mathbb{R}) \times H^1(\mathbb{R}) \to \mathbb{R}$ the billinear form associated with the Black-Scholes operator in the transformed domain, namely, $a^{BSM}(\phi, \psi) = (\phi', \psi')$. Let $K := (0, T]$ and $G := (-\infty, \log(B))$.

The variational formulation of our problem is as follows:

$$\begin{aligned} &\text{find} \quad u \in L^2(K; H^1(G)) \cap H^1(K; L^2(G)) \quad \text{s.t.} \\ &(\partial_t u, v) = a^{BSM}(u, v), \quad \forall v \in H^1(G), \quad \text{a.e. in } K \\ &\text{where} \quad u(0, x) = u_0(x) \quad \text{and} \quad u|_{\partial G} = 0 \end{aligned} \tag{4.15}$$

In our case $u_0(x) \in L^2(G)$, and, therefore, a unique solution is guaranteed. This is because $a^{BSM}(\cdot,\cdot)$ is continuous and satisfies a Garding inequality on $H^1(G)$. For our multilevel discretiza-

tion, we define a family of nested simplices $\{\Gamma_j\}_{j=0}^{J}$, with $\frac{1}{c}2^{j-J} \leq h_j = \max_{T\in\Gamma_j} \text{diam}(T) \leq c2^{j-J}$. We take our finite element spaces to be

$$V_j = \{u \in C(G) : u|_{\partial G} = 0, u|_T \in P_1(T), \forall\ T \in \Gamma_j\},$$

where $P_1(T)$ denotes the linear functions on the simplex T. We have

$$V_J \subset V_{J-1} \subset \cdots \subset V_0 \subset H^1(G).$$

This gives us finite element approximations on each level, represented as follows:

$$\text{find } u_j \in K \times V_j \text{ such that } a^{BSM}(u_j, v_j) = (\partial_t u_j, v_j), \quad \forall\ v_j \in V_j \tag{4.16}$$

We note that in the case of Black-Scholes operator in the transformed domain with an evenly spaced discretization, we will end up with a matrix equivalent to the finite differences approach. However, for tmore complex models this is not the case.

4.4 Local Mode Analysis

We consider the application of local Fourier analysis (LFA) to our discrete problem. We aim to produce an anisotropy ratio $\lambda_a := h_t/h_x^2$ to determine whether coarsening in space or time produces a preferable convergence rate on a given grid level. The work in this section follows from [23, 41, 46]. We adopt the notation from [23].

We note that we are unable to perform rigorous Fourier analysis for the majority of cases because it requires the existence of an orthogonal basis of periodic eigenfunctions for the operator. Rigorous Fourier analysis can only be applied to a very small class of problems in practice [47]. It is for this reason we turn to local Fourier analysis. This technique was first introduced by Brandt, in [6]. The analysis is performed locally, and assumes our problem to be a linear differential equation with constant coefficients on an infinite grid. In general, the equations dealt with in option pricing do not fit this criteria, but locally can be assumed to be of this form. However, problems can occur when attempting to produce a result for an entire grid. In this case, a suitable freezing of coefficients must be chosen. In Section 4.5 we discuss the forms of non-constant coefficient parabolic PDEs that are well suited to this analysis and, more importantly, our technique. In addition, local Fourier analysis assumes the boundary conditions to be periodic in nature. This is never the case for option valuation, specifically at $t = 0$, but we suppose our intial condition to be sufficiently smooth.

We begin by detailing the local Fourier analysis technique in two dimensions. We aim to approximate the spectral matrix of the two-grid operator

$$M_h^H = S_h^{\nu_2}(I_h - I_H^h L_H^{-1} I_h^H F_h L_h) S_h^{\nu_1} \tag{4.17}$$

where S_h is a given smoothing operator, ν_1 and ν_2 are the number of pre- and post- smothing

iterations performed, I_h is the identity operator, and I_h^H and I_H^h are the restriction and prolongation operators, respectively. We use F_h as a normalizing constant; we have $F_h = 1$ for space coarsening and $F_h = 2$ for time and simultaneous coarsening. We attempt to approximate M_h^H using an operator $\hat{M}_h^H$ defined on the frequency domain of our problem. We define the frequency domain operator $\hat{M}_h^H$ in similar way:

$$\hat{M}_h^H = \hat{S}_h^{\nu_2}(\hat{I}_h - \hat{I}_H^h \hat{L}_H^{-1} \hat{I}_h^H F_h \hat{L}_h)\hat{S}_h^{\nu_1}. \tag{4.18}$$

Each of these operators is represented by a 4×4 matrix acting on the frequency domain $[-\pi, \pi)^2$. We note that for the operators we are considering, we have $(I_h^H)^T = I_H^h$ and $(\hat{I}_h^H)^T = \hat{I}_H^h$; however, this is not always the case.

We define $\Theta_h = \{(\theta_1, \theta_2) : \theta_\alpha = 2\Pi k_\alpha / n_\alpha\}$, $k_\alpha = -n_\alpha/2 + 1, ..., n_\alpha/2$ ($k_1 = M$, $k_2 = N$) to be a discretization of the frequency domain. Given some choice $\theta^1 \in \Theta_h \cap [-\pi/2, \pi/2)^2$, we define the following vectors.

$$\theta^2 = \theta^1 - \begin{pmatrix} sign(\theta_1^1)\pi \\ sign(\theta_2^1)\pi \end{pmatrix} \tag{4.19}$$

$$\theta^3 = \theta^1 - \begin{pmatrix} 0 \\ sign(\theta_2^1)\pi \end{pmatrix} \tag{4.20}$$

$$\theta^4 = \theta^1 - \begin{pmatrix} sign(\theta_1^1)\pi \\ 0 \end{pmatrix} \tag{4.21}$$

We introduce exponential Fourier modes $\psi_h(\theta)_j = e^{ij\cdot\theta}$, $j = (j_1, j_2)$, $j_\alpha = 0, ..., n_\alpha - 1$. For $\theta \in \Theta_h \cap [-\pi/2, \pi/2)^2$, we define $\Psi_h(\theta) = (\psi_h(\theta^1), \psi_h(\theta^2), \psi_h(\theta^3), \psi_h(\theta^4))^T$. We note that the linear space spanned by $\Psi_h(\theta)$ is invariant under the two-grid operator $\hat{M}_h^H(\theta)$. We see that when $\psi_h(\theta)$ is projected to the coarse grid, it aliases with $\psi_H(\bar{\theta})$, where $\bar{\theta}$ equals $(2\theta_1, \theta_2)$ for coarsening in space, $(\theta_1, 2\theta_2)$ for coarsening in time, and $(2\theta_1, 2\theta_2)$ for simultaneous coarsening [23]. We give Figures 4.3, 4.4, and 4.5, taken from [43], to illustrate this for the three different cases.

All that remains is to consider the representations for $\hat{L}_h$, $\hat{L}_H$, $\hat{I}_h^H$, and $\hat{S}_h$. Let l_k, s_k, r_k, and p_k represent the stencils for L_h, L_H, I_H^h and I_h^H, respectively, where k ranges over an index set $J \subset \mathbb{Z}^2$. We begin with $\hat{L}_h$. Its representation is given by

$$\hat{L}_h(\theta) = \begin{pmatrix} \tilde{L}_h(\theta^1) & 0 & 0 & 0 \\ 0 & \tilde{L}_h(\theta^2) & 0 & 0 \\ 0 & 0 & \tilde{L}_h(\theta^3) & 0 \\ 0 & 0 & 0 & \tilde{L}_h(\theta^4) \end{pmatrix}, \tag{4.22}$$

where $\tilde{L}_h(\theta) = \sum_{k\in J} l_k e^{ik\cdot\theta}$. For the coarse grid $\hat{L}_H$, we have

$$\hat{L}_H(\theta) = \begin{pmatrix} \tilde{L}_H(\bar{\theta}^1) & 0 \\ 0 & \tilde{L}_H(\bar{\theta}^3) \end{pmatrix}, \begin{pmatrix} \tilde{L}_H(\bar{\theta}^1) & 0 \\ 0 & \tilde{L}_H(\bar{\theta}^4) \end{pmatrix}, \qquad \left(\tilde{L}_H(\bar{\theta}^1)\right) \tag{4.23}$$

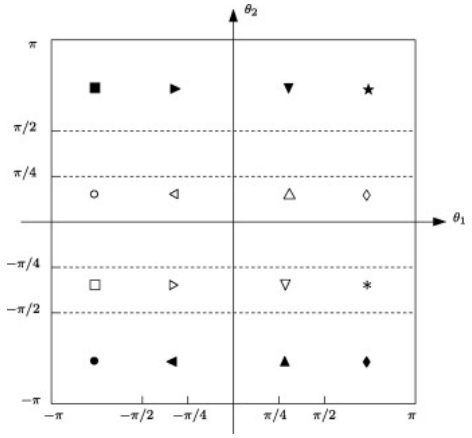


Figure 4.3. Aliasing Fourier Modes for Coarsening in Space

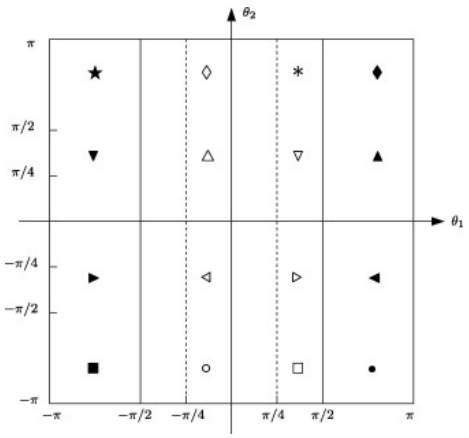


Figure 4.4. Aliasing Fourier Modes for Coarsening in Time

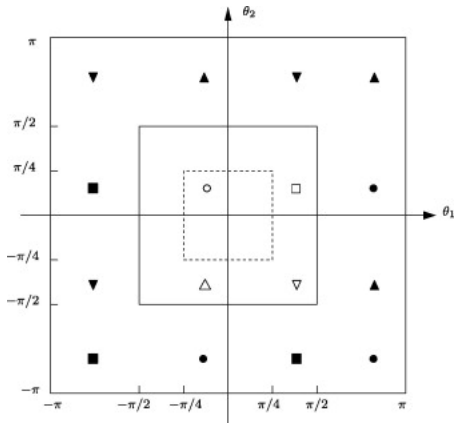


Figure 4.5. Aliasing Fourier Modes for Simultaneous Space-Time Coarsening

for coarsening in space, time, and space and time, respectively. Similarly to $\tilde{L}_h(\theta)$, we have $\tilde{L}_H(\theta) = \sum_{k\in J} s_k e^{ik\cdot\theta}$. The prologation operator $\hat{I}_H^h(\theta)$ is represented by

$$\hat{I}_H^h(\theta) = \begin{pmatrix} \tilde{I}_H^h(\theta^1) & 0 \\ 0 & \tilde{I}_H^h(\theta^2) \\ 0 & \tilde{I}_H^h(\theta^3) \\ \tilde{I}_H^h(\theta^4) & 0 \end{pmatrix}, \quad \begin{pmatrix} \tilde{I}_H^h(\theta^1) & 0 \\ 0 & \tilde{I}_H^h(\theta^2) \\ \tilde{I}_H^h(\theta^3) & 0 \\ 0 & \tilde{I}_H^h(\theta^4) \end{pmatrix}, \quad \begin{pmatrix} \tilde{I}_H^h(\theta^1) \\ \tilde{I}_H^h(\theta^2) \\ \tilde{I}_H^h(\theta^3) \\ \tilde{I}_H^h(\theta^4) \end{pmatrix} \tag{4.24}$$

for coarsening in space, time, and space and time, respectively. We have $\tilde{I}_H^h$ given by $\tilde{I}_H^h(\theta) =$

$\frac{h_1 h_2}{H_1 H_2} \sum_{k \in J} p_{-k} e^{ik \cdot \theta}$. The operator $\hat{S}_h(\theta)$ for Gauss-Seidel red-black smoothing has the form

$$\hat{S}_h(\theta) = \frac{1}{2} \begin{pmatrix} \alpha(\theta^1) + \beta(\theta^1) & \alpha(\theta^2) - \beta(\theta^2) & 0 & 0 \\ \alpha(\theta^1) - \beta(\theta^1) & \alpha(\theta^2) + \beta(\theta^2) & 0 & 0 \\ 0 & 0 & \alpha(\theta^3) + \beta(\theta^3) & \alpha(\theta^4) - \beta(\theta^4) \\ 0 & 0 & \alpha(\theta^3) - \beta(\theta^3) & \alpha(\theta^4) + \beta(\theta^4) \end{pmatrix}, \tag{4.25}$$

where the functions $\alpha(\theta)$ and $\beta(\theta)$ have the representation

$$\alpha(\theta) = -\frac{1}{l_{(0,0)}} \sum_{k \in J_0} l_k e^{ik \cdot \theta} \tag{4.26}$$

$$\beta(\theta) = -\frac{1}{l_{(0,0)}} \Big(\sum_{|k| = odd} l_k \alpha(\theta) e^{ik \cdot \theta} + \sum_{0 \neq |k| = even} l_k e^{ik \cdot \theta} \Big), \tag{4.27}$$

with $J_0 = J \backslash \{(0,0)\}$ and $|k| = |k_1| + |k_2|$. We note that the interpolation elements are independant of the specific parabolic PDE and choice of numerical ODE method. We give the following representations for the prolongation elements defined in Section 4.3.

$$(\tilde{I}_H^h)_s(\theta) = \frac{1}{2}(1 + \cos(\theta_1)) \tag{4.28}$$

$$(\tilde{I}_H^h)_t(\theta) = \frac{1}{2}(1 + e^{-i\theta_2}) \tag{4.29}$$

$$(\tilde{I}_H^h)_{st}(\theta) = \frac{1}{4}(1 + \cos(\theta_1))(1 + e^{-i\theta_2}) \tag{4.30}$$

What remains to be considered are the representations for the elements of the PDE itself and the smoother, which vary by equation and discretization. For the equation $u_t = u_{xx}$, the exact formulae have been studied in [23] for a number of different methods, including the implicit Euler and Crank-Nicolson methods. We include the results for completeness. For the implicit Euler method we have

$$\tilde{L}_h(\theta) = 1 - e^{-i\theta_2} + 2\lambda(1 - \cos(\theta_1)) \tag{4.31}$$

$$\alpha(\theta) = \frac{e^{-i\theta_2} + 2\lambda \cos(\theta_1)}{1 + 2\lambda} \tag{4.32}$$

$$\beta(\theta) = \alpha^2(\theta), \tag{4.33}$$

and for the Crank-Nicolson method,

$$\tilde{L}_h(\theta) = 1 - e^{-i\theta_2} + \lambda(1 - \cos(\theta_1))(1 + e^{-i\theta_2}) \tag{4.34}$$

$$\alpha(\theta) = \frac{(1 - \lambda)e^{-i\theta_2} + \lambda \cos(\theta_1)(1 + e^{-i\theta_2})}{1 + \lambda} \tag{4.35}$$

$$\beta(\theta) = \frac{(1 - \lambda)\alpha(\theta)e^{-i\theta_2} + \lambda \cos(\theta_1)(\alpha(\theta) + e^{-i\theta_2})}{1 + \lambda}. \tag{4.36}$$

4.4.1 Numerical Results

We consider the numerical application of our multigrid method, using local Fourier analysis to determine our restriction operator. Using the explicit representations of the stencils of the operators in the frequency domain, we can compute the spectral radius of $\hat{M}_h^H$ for a given $\theta \in \Theta_h \cap [-\pi/2, \pi/2)^2$. We define $\tilde{\rho} = \max\{\rho(\hat{M}_h^H) \mid \theta \in \Theta_h \cap [-\pi/2, \pi/2)^2\}$, where $\rho(\cdot)$ is the spectral radius. This gives an indicative measure of the convergence rate for a given choice of λ_a and restriction operator. We note the simiplicity in which this computation is done in practice. The majority of the legwork is done by hand, and, once completed, requires a nominal amount of computing time, in comparison to the multigrid iteration itself.

For numerical illustration, we consider the problem of pricing an up-and-out barrier option with the following conditions.

$$B = 20,\ K = 13,\ T = 1,\ r = 0.1,\ \sigma = 0.25 \tag{4.37}$$

For the test problem (4.37), we computed values of $\tilde{\rho}$ for a range of values of λ. We consider the value of λ for which $\tilde{\rho}_x \approx \tilde{\rho}_t$. We note that for the implicit Euler method this value is clear and well-defined, but for the Crank-Nicolson method, which exhibits more erratic behavior with respect to λ, there are a number of such values. For this case, the choice of λ_a is somewhat qualitative. We take $\lambda_a = 2^{-3/4}$ for the implicit Euler method and $\lambda_a = 2^{-1}$ for the Crank-Nicolson method. It is these values that are used as the cutoff for coarsening in space and in time for our technique.

We consider convergence results for the given test problem. Although we have only detailed the applicability of our technique to the transformed domain, we compute numerical tests for both the standard and transformed Black-Scholes PDE. For the standard domain, we freeze x at $B/\sqrt{2}$. For the choice of cut-off in the standard domain, it suffices to divide the value of λ_a for the transformed domain by $\frac{\sigma^2 B^2}{4} = \frac{25}{4} \approx 2^{11/4}$. This can be justified by considering $u_t = \frac{1}{2}\sigma^2 x^2 u_{xx}$ and noting the nominal effect lower order terms play in LFA.

We compute the approximate convergence rates for different grid sizes, for both the standard and transformed domain. We use $\rho = \left(\frac{d^{(n)}}{d^{(0)}}\right)^{(1/n)}$ as a measure of convergence, where $d^{(i)} = f - Au^{(i)}$ and n is the number of iterations for $d^{(i)}$ to satisfy the given tolerance $||d^{(i)}|| < c$. For our tests, we take $c = 10^{-12}$. We have the results in Table 4.3.

Table 4.3. Convergence Factors for Adaptive Space-Time V-Cycle(2,2)

		Standard Domain			Transformed Domain		
Method	M \ N	33	65	129	33	65	129
Implicit Euler	33	.1629	.4355	.6673	.0175	.0251	.1189
	65	.1889	.4339	.6635	.0608	.0745	.1855
	129	.2887	.5390	.6970	.2444	.1988	.2598
Crank-Nicolson	33	.0386	.1439	.3339	.0090	.0123	.0376
	65	.0946	.1705	.3215	.0540	.0533	.0728
	129	.1921	.2585	.3611	.2555	.1933	.2186

We see immediately that the convergence rates in the transformed domain are superior to that of the standard domain. This can be attributed to the fact that local Fourier analysis results for the standard domain are not necessarily applicable. We see that the extra assumption of a constant coefficient operator, especially in the terms of higher differential order, is one that greatly affects the results. We stress this trend for our method, and will give more rigorous justification in Section 4.5.

In general, even the results of the transformed domain are slightly higher than the results obtained for the heat equation in [23]. This can be attributed to the non-periodic initial condition, and will be observed to some extent when any technique is applied to a problem with such initial conditions.

4.5 Local Fourier Analysis for Parabolic Equations in $\mathbb{R}^d$

It can be seen that the process of local Fourier analysis does not apply well to partial differential equations with non-constant coefficients in the higher differential order terms. Although the ideal formulation is of the heat equation in some transformed variables, as done in Section 4.2, this is not always possible in practice. The form $u_t = \Delta u + \delta(x)u$ is the next best choice, and is still significantly more suitable to local Fourier analysis than the general non-constant coefficient linear parabolic equation. For the equation $u_t = x^\alpha u_{xx}$ the amount of perturbation of the results of LFA by changes in x are unbounded. However, under suitable conditions, for the form $u_t = \Delta u + \delta(x)u$ we can put strict bounds on the pertubation of the two-grid analysis. We stress the importance of two-grid convergence results that are robust to the entire domain. Before giving such results, we must give the formulation of local mode analysis for parabolic equations in $\mathbb{R}^d$. The setup follows very closely to the work in Section 4.4.

We consider the frequency domain $[-\pi, \pi)^{d+1}$, coupled with some discretization $\Theta_h = \{\theta : \theta_\alpha = 2\pi k_\alpha/n_\alpha, k_\alpha = -n_\alpha/2 + 1, ..., n_\alpha/2\}$. We assume θ_{d+1} to be the frequency component of the time variable. Given a choice $\theta \in \Theta_h \cap [-\pi/2, \pi/2)^{d+1}$, we can define 2^{d+1} vectors $\theta^1, \theta^2, ..., \theta^{2^{d+1}}$ by

$$\theta^m = \theta - \Gamma,$$

where $\Gamma_i := \mathbf{1}_{i \in \mathcal{C}_m} sign(\theta_i)\pi$, and $\{\mathcal{C}_i\}_1^{2^{d+1}}$ are the 2^{d+1} different combinations of the set $\mathbb{Z}_{d+1} = \{1, ..., d+1\}$. Therefore, each operator in the frequency domain is represented by a $2^{d+1} \times 2^{d+1}$ matrix. The exponential Fourier mode $\phi_h(\theta)$ aliases with $\phi_H(\bar{\theta})$ upon restriction, with $\bar{\theta}$ equal to $(2\theta_1, 2\theta_2, ..., 2\theta_d, \theta_{d+1})$ for coarsening in space, $(\theta_1, \theta_2, ..., \theta_d, 2\theta_{d+1})$ for coarsening in time, and $(2\theta_1, 2\theta_2, ..., 2\theta_d, 2\theta_{d+1})$ for simultaneous coarsening. The extensions of the operators from the case of $\mathbb{R}^2$ to $\mathbb{R}^{d+1}$ are natural and omitted. However, we note that the interpolation operators are dependent on the ordering of the combinations $\{\mathcal{C}_i\}_1^{2^{d+1}}$. We do require representations for the elements $\tilde{L}_h(\theta)$, $\alpha(\theta)$, and $\beta(\theta)$. We denote by $l_k^{\Delta+\delta}$ the stencil elements for $L_h^{\Delta+\delta}$. We begin with

the standard diffusion equation $u_t = \Delta u$, which has representation

$$\tilde{L}_h(\theta) = 1 - e^{-i\theta_{d+1}} + 2\sum_{i=1}^{d} \lambda_i(1 - \cos(\theta_i)) \tag{4.38}$$

$$\alpha(\theta) = \frac{e^{-i\theta_{d+1}} + 2\sum_{i=1}^{d} \lambda_i \cos(\theta_i)}{1 + 2\sum_{i=1}^{d} \lambda_i} \tag{4.39}$$

$$\beta(\theta) = \alpha^2(\theta) \tag{4.40}$$

for the implicit Euler method, and

$$\tilde{L}_h(\theta) = 1 - e^{-i\theta_{d+1}} + (1 + e^{-i\theta_{d+1}})\sum_{i=1}^{d} \lambda_i(1 - \cos(\theta_i)) \tag{4.41}$$

$$\alpha(\theta) = \frac{e^{-i\theta_{d+1}}(1 - \sum_{i=1}^{d} \lambda_i) + (1 + e^{-i\theta_{d+1}})\sum_{i=1}^{d} \lambda_i \cos(\theta_i)}{1 + \sum_{i=1}^{d} \lambda_i} \tag{4.42}$$

$$\beta(\theta) = \frac{\alpha(\theta)e^{-i\theta_{d+1}}(1 - \sum_{i=1}^{d} \lambda_i) + (\alpha(\theta) + e^{-i\theta_{d+1}})\sum_{i=1}^{d} \lambda_i \cos(\theta_i)}{1 + \sum_{i=1}^{d} \lambda_i} \tag{4.43}$$

for the Crank-Nicolson method. When we consider the perturbed equation $u_t = \Delta u + \delta(x)u$, we will write our elements as the sum of the δ-free terms and the extra $\delta(x)$ perturbation terms, denoting the δ perturbation with a δ superscript. We include the element $(\tilde{L}_H^{\Delta+\delta}(\theta))^{-1}$ in the same separable form as well. To do so, we must make use of the property

$$\frac{x}{y+\epsilon} = \frac{x}{y} - \epsilon\frac{x}{y(y+\epsilon)}. \tag{4.44}$$

Through use of (4.44), we can obtain the general representations in separable form.

$$\begin{aligned} \tilde{L}_h^{\Delta+\delta}(\theta) &= \sum_{k\in J} l_k^{\Delta+\delta} e^{ik\cdot\theta} = \sum_{k\in J} l_k e^{ik\cdot\theta} + \sum_{k\in J} l_k^{\delta} e^{ik\cdot\theta} \\ &= \tilde{L}_h(\theta) + \sum_{k\in J} l_k^{\delta} e^{ik\cdot\theta} \end{aligned} \tag{4.45}$$

$$\begin{aligned} (\tilde{L}_H^{\Delta+\delta}(\theta))^{-1} &= \frac{1}{\tilde{L}_H(\theta) + \tilde{L}_H^{\delta}(\theta)} \\ &= (\tilde{L}_H(\theta))^{-1} - \frac{\tilde{L}_H^{\delta}(\theta)}{\tilde{L}_H^{\Delta+\delta}(\theta)\tilde{L}_H(\theta)} \end{aligned} \tag{4.46}$$

$$\begin{aligned} \alpha_{\Delta+\delta}(\theta) &= -\frac{1}{l_{(0,0)}^{\Delta+\delta}} \sum_{k\in J_0} l_k^{\Delta+\delta} e^{ik\cdot\theta} \\ &= \alpha(\theta) - \frac{1}{l_{(0,0)}^{\Delta+\delta}} \left(\sum_{k\in J_0} l_k^{\delta} e^{ik\cdot\theta} + l_{(0,0)}^{\delta}\alpha(\theta) \right) \end{aligned} \tag{4.47}$$

$$\begin{aligned}\beta_{\Delta+\delta}(\theta) &= -\frac{1}{l_{(0,0)}^{\Delta+\delta}}\Big(\sum_{|k|=odd} l_k^{\Delta+\delta}\alpha_{\Delta+\delta}(\theta)e^{ik\cdot\theta} + \sum_{0\neq|k|=even} l_k^{\Delta+\delta}e^{ik\cdot\theta}\Big)\\ &= \beta(\theta) - \frac{1}{l_{(0,0)}^{\Delta+\delta}}\Big(\alpha(\theta)\sum_{|k|=odd} l_k^{\delta}e^{ik\cdot\theta} + \alpha_\delta(\theta)\sum_{|k|=odd} l_k^{\Delta+\delta}e^{ik\cdot\theta}\\ &+ \sum_{0\neq|k|=even} l_k^{\Delta+\delta}e^{ik\cdot\theta} + \beta(\theta)l_{(0,0)}^{\delta}\Big) \end{aligned} \tag{4.48}$$

These are independant of the choice of discretization in time. For the two methods we consider, we have the following formulae for our perturbations $\Box^\delta := \Box^{\Delta+\delta} - \Box^\Delta$.

$$ {}_{ie}\tilde{L}_h^\delta(\theta) = -h_t\delta(x) \qquad {}_{cn}\tilde{L}_h^\delta(\theta) = -\frac{1}{2}(1+e^{-i\theta_{d+1}})h_t\delta(x) \tag{4.49}$$

$$ {}_{ie}(\tilde{L}_H^\delta(\theta))^{-1} = \frac{h_t\delta(x)}{\tilde{L}_H^{\Delta+\delta}(\theta)\tilde{L}_H(\theta)} \qquad {}_{cn}\tilde{L}_H^{-\delta}(\theta) = \frac{\frac{1}{2}(1+e^{-i\theta_{d+1}})h_t\delta(x)}{\tilde{L}_H^{\Delta+\delta}(\theta)\tilde{L}_H(\theta)} \tag{4.50}$$

$$ {}_{ie}\alpha_\delta(\theta) = -\frac{h_t\delta(x)\alpha(\theta)}{1+2\sum_{i=1}^d\lambda_i - h_t\delta(x)} \qquad {}_{cn}\alpha_\delta(\theta) = -\frac{\frac{h_t}{2}\delta(x)(e^{-i\theta_{d+1}}+\alpha(\theta))}{1+\sum_{i=1}^d\lambda_i - \frac{h_t}{2}\delta(x)} \tag{4.51}$$

$$\begin{aligned} {}_{ie}\beta_\delta(\theta) = -\frac{1}{1+2\sum_{i=1}^d\lambda_i - h_t\delta(x)}\big(&\alpha_\delta(\theta)e^{-i\theta_{d+1}}(1-\sum_{i=1}^d\lambda_i)\\ &+\alpha_\delta(\theta)(1+e^{-i\theta_{d+1}})\sum_{i=1}^d\lambda_i\cos(\theta_i) - h_t\delta\beta(\theta)\big) \end{aligned} \tag{4.52}$$

$$\begin{aligned} {}_{cn}\beta_\delta(\theta) = -\frac{1}{1+\sum_{i=1}^d\lambda_i - \frac{h_t}{2}\delta(x)}\big(&-\frac{h_t}{2}\delta(x)(\alpha(\theta)e^{-i\theta_{d+1}}+\beta(\theta))\\ &+\alpha_\delta(\theta)(\alpha(\theta)(1+\sum_{i=1}^d\lambda_i) - e^{-i\theta_{d+1}}\sum_{i=1}^d\lambda_i\cos(\theta_i))\big) \end{aligned} \tag{4.53}$$

Now that we have characterized the elements and operators in the frequency domain, we are prepared to give the following theorem.

Theorem 4.5.1. *Let $\hat{M}_h^H$ and $(\hat{M}_h^H)^{\Delta+\delta}$ be the two-grid frequency domain operators of the method of lines discretization (using implicit Euler or Crank-Nicolson) of $u_t = \Delta u$ and $u_t = \Delta u + \delta(x)u$, respectively. Let $\hat{E}_{2G} = \hat{I}_h - \hat{I}_H^h\hat{L}_H^{-1}\hat{I}_h^H F_h\hat{L}_h$, $\Lambda = \sum_{i=1}^d\lambda_i$, $\epsilon = \min_{\theta\in\Theta_h}\tilde{L}_h(\theta)$, and $\Xi = \max_{\theta\in\Theta_h}\|\hat{E}_{2G}\|$. Suppose that Θ_h is bounded away from a neighborhood of zero. In addition, suppose that h_t is sufficiently small, and that a smoothing property holds, namely, $h_t\delta(x) < \min(\frac{\epsilon}{2},\Lambda)$ and $\alpha(\theta) < 1$. Then we have*

$$\max_{\theta\in\Theta_h}|\rho[\hat{M}_h^H(\theta)] - \rho[(\hat{M}_h^H)^{\Delta+\delta}(\theta;x)]| \leq Ch_t\delta(x), \tag{4.54}$$

with $C = \frac{4(\nu_1+\nu_2)}{1+\Lambda/2}\Xi + \frac{1+5(1+\Xi)}{F_h\epsilon} + O(h_t\delta(x))$.

Proof. Recall that $\hat{M}_h^H = \hat{S}_h^{\nu_2}(\hat{I}_h - \hat{I}_H^h\hat{L}_H^{-1}\hat{I}_h^H F_h\hat{L}_h)\hat{S}_h^{\nu_1}$. From the structure of the equation $u_t = \Delta u + \delta(x)u$, we can treat it as a perturbation of the d-dimensional heat equation. We will follow

this trend in the representation for $(\hat{M}_h^H)^{\Delta+\delta}$. It can be represented as

$$(\hat{M}_h^H)^{\Delta+\delta} = (\hat{S}_h + \hat{S}_h^\delta)^{\nu_2}[\hat{I}_h - \hat{I}_H^h(\hat{L}_H^{-1} + \hat{L}_H^{-\delta})\hat{I}_h^H F_h(\hat{L}_h + \hat{L}_h^\delta)](\hat{S}_h + \hat{S}_h^\delta)^{\nu_1}.$$

Expanding $(\hat{M}_h^H)^{\Delta+\delta}$, and considering the difference between $\hat{M}_h^H$ and $(\hat{M}_h^H)^{\Delta+\delta}$, we have

$$\begin{aligned}(\hat{M}_h^H)^{\Delta+\delta} - \hat{M}_h^H = [(\hat{S}_h + \hat{S}_h)^{\nu_2} - \hat{S}_h^{\nu_2}][\hat{E}_{2G} - F_h(\hat{I}_H^h\hat{L}_H^{-1}\hat{I}_h^H\hat{L}_h^\delta + \hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h \\ + \hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h^\delta)](\hat{S}_h + \hat{S}_h^\delta)^{\nu_1} + \hat{S}_h^{\nu_2}\hat{E}_{2G}[(\hat{S}_h + \hat{S}_h^\delta)^{\nu_1} - \hat{S}_h^{\nu_1}] \\ - \hat{S}_h^{\nu_2}F_h(\hat{I}_H^h\hat{L}_H^{-1}\hat{I}_h^H\hat{L}_h^\delta + \hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h + \hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h^\delta)(\hat{S}_h + \hat{S}_h^\delta)^{\nu_1}.\end{aligned}$$

Taking the norms of both sides and expanding gives us the bound

$$\begin{aligned}\|(\hat{M}_h^H)^{\Delta+\delta} - \hat{M}_h^H\| \leq (\nu_2\|\hat{S}_h^\delta\| + O(\|\hat{S}_h^\delta\|^2))(\|\hat{E}_{2G}\| + F_h(\|\hat{L}_H^{-1}\|\|\hat{L}_h^\delta\| + \|\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h\| \\ + \|\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h^\delta\|))(1 + \nu_1\|\hat{S}_h^\delta\| + O(\|\hat{S}_h^\delta\|^2)) + F_h(\|\hat{L}_H^{-1}\|\|\hat{L}_h^\delta\| \\ + \|\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h\| + \|\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h^\delta\|)(1 + \nu_1\|\hat{S}_h^\delta\| + O(\|\hat{S}_h^\delta\|^2)) \\ + \|\hat{E}_{2G}\|(\nu_1\|\hat{S}_h^\delta\| + O(\|\hat{S}_h^\delta\|^2))\end{aligned}$$

We need bounds on a number of operators. Making use of the conditions of the theorem, we have $\|\hat{E}_{2G}\| < \Xi$ and $\|\hat{L}_H^{-1}\| < \frac{1}{\epsilon}$. By inspection of (4.49), we immediately see that $\|\hat{L}_h^\delta\| < h_t\delta$. What remains is to obtain bounds on $\|\hat{S}_h^\delta\|$, $\|\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h\|$, and $\|\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h^\delta\|$.

We begin with $\|\hat{S}_h\|$. We note that the smoothing property $\alpha < 1$ implies $\beta < 1$. Using the assumption of $h_t\delta < \Lambda$, we can deduce from results (4.51,4.52,4.53) that ${}_{ie}\alpha_\delta < \frac{h_t\delta}{1+\Lambda}$ and ${}_{ie}\beta_\delta < \frac{3h_t\delta}{1+\Lambda}$, ${}_{cn}\alpha_\delta < \frac{h_t\delta}{1+\Lambda/2}$ and ${}_{cn}\beta_\delta < \frac{3h_t\delta}{1+\Lambda/2}$, and, therefore, $\|\hat{S}_h^\delta\| < \frac{4h_t\delta}{1+\Lambda/2}$.

Moving on to $\|\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h\|$, we start by observing that

$$\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h = F_h^{-1}\hat{I}_H^h\hat{L}_H^{-\delta}\hat{L}_H\hat{I}_h^H[\hat{I}_H^h L_H^{-1}\hat{I}_h^H F_h\hat{L}_h] = F_h^{-1}\hat{I}_H^h\hat{L}_H^{-\delta}\hat{L}_H\hat{I}_h^H[\hat{I}_h - \hat{E}_{2G}].$$

This gives us

$$\begin{aligned}\|\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h\| < F_h^{-1}(1+\Xi)\|\hat{I}_H^h\hat{L}_H^{-\delta}\hat{L}_H\hat{I}_h^H\| < F_h^{-1}(1+\Xi)\|\hat{L}_H^{-\delta}\hat{L}_H\| \\ < \frac{F_h^{-1}(1+\Xi)h_t\delta}{\epsilon - h_t\delta} < (1+\Xi)\frac{2F_h^{-1}}{\epsilon}h_t\delta.\end{aligned}$$

Finding a bound for $\|\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h^\delta\|$ follows a similar process. We have

$$\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h^\delta = \hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h^{\Delta+\delta} - \hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h.$$

We have a bound for the second term, and the bound for the first term is half of the first, using the same technique. This gives us $\|\hat{I}_H^h\hat{L}_H^{-\delta}\hat{I}_h^H\hat{L}_h^\delta\| < (1+\Xi)\frac{3F_h^{-1}}{\epsilon}h_t\delta$.

Making use of these bounds and noting that $|\rho[\hat{M}_h^H] - \rho[(\hat{M}_h^H)^{\Delta+\delta}]| \leq \|\hat{M}_h^H - (\hat{M}_h^H)^{\Delta+\delta}\|_2$, we

obtain

$$|\rho[\hat{M}_h^H] - \rho[(\hat{M}_h^H)^{\Delta+\delta}]| \leq (\frac{4\nu_2 h_t\delta}{1+\Lambda/2} + O(h_t^2\delta^2))(\Xi + \frac{F_h^{-1}}{\epsilon}(1+5(1+\Xi))h_t\delta(1+\frac{4\nu_1 h_t\delta}{1+\Lambda/2}$$
$$+ O(h_t^2\delta^2)) + \frac{F_h^{-1}}{\epsilon}(1+5(1+\Xi))h_t\delta(1+\frac{4\nu_1 h_t\delta}{1+\Lambda/2} + O(h_t^2\delta^2))$$
$$+ \Xi(\frac{4\nu_1 h_t\delta}{1+\Lambda/2} + O(h_t^2\delta^2))$$

Combining all terms of order $O(h_t^2\delta^2)$ and higher, we obtain the desired result.

$$|\rho[\hat{M}_h^H] - \rho[(\hat{M}_h^H)^{\Delta+\delta}]| \leq [\frac{4(\nu_1+\nu_2)}{1+\Lambda/2}\Xi + \frac{1+5(1+\Xi)}{F_h\epsilon} + O(h_t\delta)]h_t\delta$$

□

It is noted that the above theorem and proof are somewhat technical in nature; however, the intuition gained is that the difference in the results of local mode analysis is small when $\delta(x) \ll h_t$. Therefore, for a reasonably fine grid, the results of local mode analysis for a given freezing of coefficients is fairly robust for the entire grid.

In addition, we note that the terms ϵ and Ξ are all $O(1)$ and are of reasonable value when the grid in the frequency domain is sufficiently far away from zero. Our operator $\hat{L}_h$ is degenerate at zero, but for a rough mesh that avoids zero in the frequency domain, ϵ is large enough. Assuming $n = n_\alpha$ for all α, n odd, one can compute the bound $\epsilon \geq \frac{\pi^2}{n^2}(1-\frac{\pi^2}{n^2})\Lambda$ by Taylor expansion of the cosine function.

We see that in the case of a PDE which cannot be converted to the classical form $u_t = \Delta u$, we will settle for the form $u_t = \Delta u + \delta(x)u$. We note that a small amount of variance in our two-grid convergence results is acceptable and affects the cutoff nominally. This allows us to apply our method to a larger class of models and be assured that our model will perform well. In particular, this allows us to consider more general models for pricing options.

4.6 Non-Constant Volatility Models

In practice, it is known that the Black-Scholes model is far from practical, and does not give a true representation of the behavior of the volatility. The model fails to capture the so-called volatility "smile". We consider two approaches to overcome the model's shortcomings, namely, deterministic and surface volatility models.

The deterministic volatility approach involves modeling the volatility as a deterministic function of the underlying. The volatility function is calibrated so that it accurately captures the volatility smile. These models are sometimes referred to as one-factor models, stemming from the single source of randomness. Advantages of such an approach is that it produces functions that are monotonic with respect to the underlying (for vanilla options), perfectly correlated with the underlying, and replicable [9]. An example of such an approach is the constant elasticity of variance (CEV) model.

The surface volatility approach takes volatility as an additional state variable, driven by an additional random noise that is correlated with that of the underlying. We note that in this case the risk neutral measure is not uniquely specified. Therefore, unlike the case of a deterministic volatility model, a risk premium must be specified to recover a pricing function. The most famous example of the surface volatility approach is Heston's model.

In this section, we consider the constant elasticity of variance (CEV) model, of which the Black-Scholes model is a special case, and Heston's model. In addition, we detail the process in which our method would be applied.

4.6.1 Constant Elasticity of Variance Model

In the CEV model we assume our underlying to be given by the following stochastic differential equation.

$$dS(t) = rS(t)\,dt + \sigma S^{\beta}(t)\,d\tilde{W}(t) \tag{4.55}$$

Following a similar process as in Section 4.2, it can be shown that the fair price of an up-and-out European call option is given by the solution of the partial differential equation

$$\begin{cases} u_t = \mathcal{A}^{CEV} - ru \quad on \quad (0,B)\times(0,T] \\ u(0,t) = u(B,t) = 0 \quad for \quad t \in [0,T) \\ u(x,0) = |x-K|_+ \qquad for \quad x \in [0,B], \end{cases} \tag{4.56}$$

with $\mathcal{A}^{CEV} = \frac{1}{2}\sigma^2 x^{2\beta}\partial_{xx} + rx\partial_x$. We see that this gives a non-constant coefficient term in front of the second derivative. As discussed in Section 4.5, this can lead to problems in local Fourier analysis, similar to the case of the Black-Scholes model. For this reason, we consider the following change of variables. Let y be given by $y = \phi(x) := \frac{\sqrt{2}}{1-\beta}x^{1-\beta}$. Then our PDE takes the form

$$u_t = u_{yy} + \left[\frac{r\sigma}{\sqrt{2}}\left(\frac{\sqrt{2}}{(1-\beta)y}\right)^{\frac{\beta}{\beta-1}} - \frac{\beta\sigma}{(1-\beta)y}\right]u_y - ru. \tag{4.57}$$

Taking $u(y,t) = e^{\psi(y)}v(y,t)$, with $\psi(y)$ given by

$$\psi(y) = \frac{\beta\sigma}{4}\left[\left(\frac{\sqrt{2}}{(1-\beta)y}\right)^{\frac{2\beta-1}{\beta-1}} - \frac{2}{(1-\beta)y^2}\right], \tag{4.58}$$

we obtain the desired form of our equation.

$$v_t = v_{yy} + (\psi''(y) + \tfrac{1}{2}(\psi'(y))^2 - r)v \tag{4.59}$$

The boundary conditions, and exact domain depend on the value of β chosen. We now have a formulation that is better suited to local Fourier analysis. However, due to the problems that occur as x approaches zero, in the numerical formulation we must create an artificial boundary at $x = \epsilon$, for some ϵ relatively small (with respect to the range of values that are of interest). The results

of local Fourier analysis are fairly robust to this case, granted that h_t is chosen to be sufficiently small.

In terms of the application of our technique, it follows closely to the Black-Scholes model, with the same interpolation operators and smoother. The difference is a slight change in the operator and initial condition. However, these differences are nominal and require very little changes.

4.6.2 Heston's Model

For Heston's model, taking volatility in this case be a martingale, the underlying follows a two-parameter stochastic differential equation, given by

$$\begin{aligned} dS_t &= rS_t dt + \sqrt{\nu_t} S_t d\tilde{W}_t^{(1)} \\ d\nu_t &= \sigma \nu_t d\tilde{W}_t^{(2)}, \end{aligned} \tag{4.60}$$

where $\mathbb{E}[d\tilde{W}_t^{(1)} d\tilde{W}_t^{(2)}] = \rho dt$ for some constant $\rho \in [-1, 1]$. Once again, by a similar process as in the Black-Scholes and CEV model, we can obtain a PDE that the fair value of our option satisifes, given by

$$\begin{cases} u_t = \mathcal{A}^H - ru & on \quad (0,B) \times (0,\infty) \times [0,T) \\ u(0,\nu,t) = u(B,\nu,t) = 0 & for \quad \nu \in (0,\infty),\ t \in [0,T) \\ u(S,\nu,T) = |S-K|_+ & for \quad S \in [0,B],\ \nu \in (0,\infty), \end{cases} \tag{4.61}$$

where $\mathcal{A}^H = \frac{1}{2}\nu\big(S^2\partial_{SS} + 2\rho\sigma S\partial_{S\nu} + \sigma^2\partial_{\nu\nu}\big) + rS\partial_S$.

Following the work done in [11], we can perform a number of coordinate transformations to obtain the form $u_t = \Delta u$. For the sake of brevity, we will not give the details of the computation, but merely the resulting coordinate transformation. We give the transformation $(S, \nu, t) \to (\gamma, \psi, \phi)$, given by

$$\gamma = \log S - \frac{\rho\nu}{\sigma} + r(T-t) \tag{4.62}$$

$$\psi = \frac{\nu}{\sigma}\sqrt{1-\rho^2} \tag{4.63}$$

$$\phi = \nu(T-t), \tag{4.64}$$

and the change of function $u(S,\nu,t) = \exp\{-r(T-t) + \frac{\phi}{4(1-\rho^2)} + \frac{\gamma}{2(1-\rho^2)}\}v(\gamma,\psi,\phi)$, resulting in the PDE

$$v_\phi = (1-\rho^2)(v_{\gamma\gamma} + v_{\psi\psi}) \tag{4.65}$$

$$v(\gamma,\psi,0) = e^{-\frac{\gamma}{2(1-\rho^2)}}\left|e^{\gamma+\rho\psi(1-\rho^2)^{-1/2}} - K\right|_+ \tag{4.66}$$

$$\gamma \in (-\infty,\infty), \quad \psi \in [0,\infty), \quad \phi \in [0,\infty). \tag{4.67}$$

From here our method can be easily applied. Some of the considerations that arise is the type of stencil used in space, how to discretize in space, and the choice of spatial interpolation. Finite

element comes into its own for two- and three- dimensional elliptic problems, and is a reasonable choice here. These choices are left to the reader.

4.7 General Discussion of Technique

Our space-time multigrid technique has both its advantages and disadvantages. We believe the main source of appeal for our algorithm, as well as its drawbacks, to be as follows:

Advantages:

- multigrid techniques are iterative in nature
- space-time multigrid, with a suitable smoother, is fully parallelizable
- the transformation in domain produces superior convergence rates, and the augmentation of the boundary allows standard theory to guarantee approximation quality
- grid spacing focused on the lower boundary, produced by the transformation of the domain minimizes problems produced by a lack of regularity at the boundary
- the method is robust to a large class of option pricing models

Disadvantages:

- treating space and time simultaneously produces large matrices.

The advantage of an iterative technique is two-fold. First, the ability to choose a stopping criterion allows the individual to decide whether to favor approximation quality or quicker computations, depending on individual needs. Secondly, it allows the use of a solution to a similar problem as an initial guess, which, for mildly perturbed problems, converges extremely quickly. When such computations are performed a large number of times, we have very good intial guesses to use. There is no need to use a non-iterative method when it is more advantageous to use pre-existing data.

With the same concept of a large number of computations of this form being done, we note that our method is also fully parallelizable, given a suitable pointwise smoother. In particular, assuming an unlimited number of processors, our method would have parallel complexity $O(\log M + \log N)$ (M and N being the number of grid points in the time and space directions, respectively) [23], as compared to time-stepping with multigrid in space, which has $O(M \log N)$ parallel complexity.

We have also shown, through numerics in Section 4.3, that the discretization in the transformed domain produces significantly better convergence rates than discretization in the standard domain. This effect is two-fold. Firstly, we produce a formulation that is well suited to local mode analysis. Moreover, the transformations tend produce a non-uniform grid in the spatial dimension that favors the the lower boundary, where the PDE degenerates to an ODE and spatial ellipticity is lost.

In addition, in the discretization of the transformed domain, we cut off the lower boundary. This cut-off results in a domain for which our problem is uniformly elliptic in the spatial dimension.

This allows for the application of standard theoretical results with respect to approximation quality and convergence results.

We have shown our method to be robust to a large class of parabolic PDEs, through spatial transformations. In particular, we have shown our method to be applicable to both deterministic and surface volatility models. We have treated three of the most popular pricing models, in the form of the Black-Scholes, CEV, and Heston's model.

The major disadvantage of the method is the large matrix it produces. To gain a great deal of accuracy in the solution, the matrix must become large. However, we stress that this algorithm is not meant to be performed on a single processor. This technique becomes adventageous when a large number of processors are available, and a short computaiton time is of great importance. The algorithm's ability to be fully parallelized makes the large matrix less relevant, although it does remain a valid point. For a single processor, multigrid purely in space, with time-stepping, would be a more suitable procedure.

4.8 Conclusion

We have shown how an adaptive space-time multigrid technique can be applied to the pricing of barrier options. We have shown that the algorithm will produce favorable convergence rates, as shown by local Fourier analysis, and confirmed by numerical results on a test problem.

We stress that our algorithm is desirable in practice, due to the very small amount of computational complexity required, given a sufficient amount of processors. We believe this makes the technique suitable for the large scale computations performed in industry. We stress that although the process of applying the algorithm in parallel is not directly addressed, we give some basic results with respect to the complexity without proof.

In addition, the techniques applied do not need to be limited to the case of barrier options, and could easily be applied to a greater class of derivatives. The choice of barrier options was mainly because of the lack of explicit formulae for solutions for more complex models. The application of this technique, and multigrid techniques in general, to the pricing of options has not been extensively studied, and is a possible avenue of further academic pursuits in financial mathematics.

Bibliography

[1] Babuska, I.; Osborn, J. E. *Finite Element-Galerkin Approximation of the Eigenvalues and Eigenvectors of Selfadjoint Problems* Mathematics of Computation, Vol. 52, No. 186. (Apr., 1989), pp. 275-297.

[2] Barnard, Stephen T.; Simon, Horst D. *Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems*. Concurrency: Practice and Experience, Vol. 6(2), 101-117 (April 1994).

[3] Berkhin, P. *Survey of clustering data mining techniques*. Technical report, Accrue Software, San Jose, California, 2002.

[4] Bollobas, Bela. *Modern Graph Theory*. Springer. (1998) ISBN: 0387984887

[5] Bondy, Adrian; Murty, U.S.R. *Graph Theory*. Springer. (2008) ISBN: 1846289696

[6] Brandt, A. *Multi-level adaptive solutions to boundary-value problems* Math. Comp., 31 (138): 333390, 1977.

[7] Brandt, A. *Multigrid techniques: 1984 guide, with application to fluid dynamics* GMD Studien 85, GMD-AIW, Postfach 1240, D-5205 St.-Augustin, Germany, 1984.

[8] Brandt, A.; McCormick, S.; Ruge, J. *Multigrid Methods for Differential Eigenproblems*. SIAM J. Sci. Stat. Comput. , Vol. 4, No. 2, June 1983, 264-260.

[9] Corradi, V., Distaso, W. *Deterministic versus stochastic volatility* Working Paper. 2007

[10] Derman, Emanuel; Kani, Iraj. *The Ins and Outs of Barrier Options: Part 1* Derivatives Quarterly. Winter 1996.

[11] Dell'Era, Mario. *Vanilla Option Pricing in Stochastic Volatility Market Models* International Review of Applied Financial issues and Economics, Forthcoming, July 12, 2012.

[12] Diestel, Reinhard. *Graph Theory*. Springer-Verlang, New York, 2000. xiv+310 pp. ISBN: 0-387-98976-5

[13] Fiedler, Miroslav. *Algebraic Connectivity of Graphs*. Czechoslovak Mathematical Journal, 23 (98) 1973, 298-305

[14] Fiedler, Miroslav. *A Property of Eigenvectors of Nonnegative Symmetric Matrices and its Application*. Czechoslovak Mathematicial Journal, Vol. 25 (1975), No. 4, 619–633

[15] Fiduccia, C. M.; Mattheyses, R. M. *A Linear Time Heuristic for Improving Network Partitions.* Proc. 19th IEEE Design Automation Conference, 1982, 175-181.

[16] Glasserman, Paul. *Monte Carlo Methods in Financial Engineering* Springer. Stochastic Modeling and Applied Probability (v. 53). 1st ed. August 2003. ISBN: 978-0387004518

[17] Golub, Gene H.; Van Loan, Charles F. *Matrix Computations.* Johns Hopkins University Press, 3rd ed. 1996. ISBN-13: 978-0801854148

[18] Hackbusch, Wolfgang. *Elliptic Differential Equations: Theory and Numerical Treatment* Springer-Verlag, 1992. ISBN-13: 978-3540548225

[19] Hackbusch, Wolfgang. *Multi-Grid Methods and Applications* Springer-Verlang, January 1986. ISBN-13: 978-0387127613

[20] Hall, K.M. *An r-dimensional Quadratic Placement Algorithm* . Management Science 17 (1970), 219-229.

[21] Hendrickson, Bruce; Kolda, Tamara. *Graph Partitioning Models for Parallel Computing.* Parallel Computing. Volume 26, Issue 12, Nov 2000, 15191534

[22] Holzrichter, M. ; Oliveira, S. *A Graph Based Method for Generating the Fiedler Vector of Irregular Problems* . IPPS/SPDP Workshops, 978-985, 1999.

[23] Horton, G.; Vandewalle, S. *A Space-Time Multigrid Method for Parabolic Partial Differential Equations.* SIAM J. Sci. Comput. Vol. 16, No. 4, pp. 848-864, July 1995

[24] Howison, Sam. *Barrier Options* Lecture Notes. Oxford Centre for Industrial and Applied Mathematics. http://people.maths.ox.ac.uk/howison/barriers.pdf

[25] Investopedia, *Up-and-In Option* http://www.investopedia.com/terms/u/up-and-inoption.asp#axzz2EaWehfVf

[26] Karypis, George; Kumar, Vipin. *Multilevel k-way Partitioning Scheme for Irregular Graphs.* Journal of Parallel and Distributed Computing. 48 (1998), 96-129

[27] Karypis, George; Kumar, Vipin. *A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs.* SIAM Journal of Scientific Computing. Vol. 20, No. 1, 359-392

[28] Kernighan, B. W.; Lin, S. *An Efficient Heuristic Procedure for Partitioning Graphs.* Bell Sys. Tech. J., 49 (1970), 291-307.

[29] Koren, Y.; Carmel, L.; Harel, D. *Drawing Huge Graphs by Algebraic Multigrid Optimization.* Multiscale Modeling and Simulation, 1(4), 2003 pp. 645673.

[30] Lin, Sensin. *Finite Difference Schemes for Heston Model* . Master's Thesis. Oxford. June 2008

[31] Mandel, Jan; McCormick, Steve. *A Multilevel Variational Method for $Au=\lambda Bu$ on Composite Grids.* Journal of Computational Physics 80, 442-452 (1989).

[32] Neymeyr, Klaus. *Solving Mesh Eigenproblems with Multigrid Efficiently*

[33] Øksendal, Bernt. *Stochastic Differential Equations: An Introduction with Applications* . Springer; 6th Ed. September 2010. ISBN: 978-3540047582

[34] Østerby, Ole. *Five ways of reducing the Crank-Nicolson oscillations.* BIT, 43 (2003), pp. 811-822.

[35] Paige, C. C. *Accuracy and Effectiveness of the Lanczos Algorithm for the Symmetric Eigenproblem.* Linear Algebra and its Applications 34:235-258 (1980).

[36] Paige, C. C.; Saunders, M. A. *Solution to Sparse Indefinite Systems of Linear Equations.* SIAM J. Numer. Anal., 12 (1974), pp. 617-629.

[37] Pothen, Alex; Simon, Horst; Liou, Kang-Pu. *Partitioning Sparse Matrices with Eigenvectors of Graphs.* SIAM J. Matrix Anal. Appl., 11(3), 1990, 430-452

[38] Sen, A.; Deng, H.; Guha, S. *On a Graph Partitioning Probelm with Applications to VLSI Layout.*Circuits and Systems, 1991.,IEEE International Symposium on, 11-14 Jun 1991, vol. 5, 2846-2849.

[39] Shreve, Steven. *Stochastic Calculus for Finance II: Continuous-Time Models* Springer Finance. 1st ed. June 2004. ISBN: 978-0387401010

[40] Steele, J. Michael. *Stochastic Calculus and Financial Applications* Springer. Stochastic Modelling and Applied Probability (Book 45). December 2010. ISBN: 978-1441928627

[41] Trottenberg, Ulrich; Oosterlee, Cornelis; Schuller, Anton; *Multigrid* Academic Press; 1st Ed. ISBN: 978-0127010700

[42] Ugur, Omur. *Introduction to Computational Finance* Imperial College Press; December 2008. ISBN: 978-1848161924

[43] van der Vegt, J. J. W.; Rhebergen, S. *hp-Multigrid as Smoother algorithm for higher order discontinuous Galerkin discretizations of advection dominated flows: Part I. Multilevel analysis* Journal of Computational Physics. Volume 231, Issue 22, September 2012

[44] Varga, Richard S. *Matrix Iterative Analysis.* Springer-Verlang, Berlin, 2000. x+358 pp. ISBN: 3-540-66321-5

[45] Wessling, P. *An Introduction to Multigrid Methods* . R.T. Edwards, Inc.; January 2004. ISBN: 978-1930217089

[46] Wienands, Roman; Joppich, Wolfgang. *Practical Fourier Analysis for Multigrid Methods* Chapman and Hall/CRC. ISBN: 978-1584884927

[47] Zhao, Yao. *Fourier Analysis and Local Fourier Analysis for Multigrid Methods* Masters Thesis. Johannes Kepler Universität. July 2009

[48] Zvan, R.; Vetzal, K.; Forsyth, P. *PDE methods for pricing barrier options* J. Econ. Dyn. Control 24, 15631590. (2000)