

ABSTRACT

Title of dissertation: EXPEDITION IN DATA AND HARMONIC
ANALYSIS ON GRAPHS

Matthew Joseph Begué, Doctor of Philosophy, 2016

Dissertation directed by: Professor Kasso A. Okoudjou
Department of Mathematics

Professor John J. Benedetto
Department of Mathematics

The graph Laplacian operator is widely studied in spectral graph theory largely due to its importance in modern data analysis. Recently, the Fourier transform and other time-frequency operators have been defined on graphs using Laplacian eigenvalues and eigenvectors. We extend these results and prove that the translation operator to the i 'th node is invertible if and only if all eigenvectors are nonzero on the i 'th node. Because of this dependency on the support of eigenvectors we study the characteristic set of Laplacian eigenvectors. We prove that the Fiedler vector of a planar graph cannot vanish on large neighborhoods and then explicitly construct a family of non-planar graphs that do exhibit this property.

We then prove original results in modern analysis on graphs. We extend results on spectral graph wavelets to create *vertex-dynamic spectral graph wavelets* whose support depends on both scale and translation parameters. We prove that Spielman's Twice-Ramanujan graph sparsifying algorithm cannot outperform his

conjectured optimal sparsification constant. Finally, we present numerical results on *graph conditioning*, in which edges of a graph are rescaled to best approximate the complete graph and reduce average commute time.

EXPEDITION IN DATA AND HARMONIC
ANALYSIS ON GRAPHS

by

Matthew Joseph Begué

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:

Professor Kasso A. Okoudjou, Chair/Advisor

Professor John J. Benedetto, Co-chair/Co-advisor

Professor Radu V. Balan

Professor Alexander Barg

Professor Patrick M. Fitzpatrick

© Copyright by
Matthew Joseph Begué
2016

Dedication

In memory of Dolores Biron.

Acknowledgments

I send an enormous thanks to my selfless advisor Dr. Kasso A. Okoudjou. Kasso has always taken interest in my research and academic progress since my undergraduate career. He is a huge reason why I was accepted and chose to attend the University of Maryland. From day one, he was a generous and caring mentor with an inspiring work ethic and was available at any hour of the day. In addition to my advisor, I am happy to call him my friend.

I am immensely grateful to my coadvisor Dr. John J. Benedetto, the joyful patriarch of the Norbert Wiener Center for Harmonic Analysis and Applications. During my tenure as the center's Associate Director, organizing the February Fourier Talks conference and involvement in multiple publications are just some of the incredible opportunities presented to me all thanks to John.

I thank Dr. Wojciech Czaja and Dr. Radu V. Balan, the other two professors of the Norbert Wiener Center, whose unique expertise have exposed me to significant problems, techniques, and skills that have been invaluable in my academic and professional career.

Thank you Chae Clark, who for the past five years was my office-mate, springboard of ideas, resident MATLAB expert, and great pal. Thank you to the other Norbert Wiener Center graduate students, past and present, for their daily comradery including James, Matt, Travis, Tim, Alex, Paul, Weilin, Mark, Christiana, Franck, and Ariel. I also thank my other math friends Rob, Ryan, Sean, Ariella, Rebecca, Adam, and Summer, for their friendship and for enduring the written

qualifying exams together.

I give heartfelt thanks to my parents, Amy and Lou, who provided bottomless love and support. Thank you to my sister Emily for always making this funny guy laugh. Finally, thank you to my fiancée, Kelsey, for her endless supply of patience, encouragement, love, and laughter. This thesis is the end of a chapter in our journey together and I can't wait to see what is in store next.

Table of Contents

List of Notations	vi
1 Introduction	1
1.1 Linear Algebra	2
1.2 Graph Definitions	4
1.2.1 The Laplacian and other matrices associated with graphs . . .	6
1.2.2 Elementary properties of the Laplacian	9
1.3 Data graphs and common graphs	12
1.4 Outline of results	14
2 Fourier Analysis on Graphs	18
2.1 The Graph Fourier Transform	18
2.2 Graph Convolution	20
2.3 Graph Modulation	22
2.4 Graph Translation	23
3 Support of eigenvectors on graphs	37
3.1 Support and the Characteristic set of the Fiedler Vector	37
3.1.1 Construction of the barren graph, $\text{Barr}(N)$	46
3.1.2 Characteristic Vertices and graph adding	54
4 Spectral Graph Wavelets	58
4.1 Introduction to the spectral graph wavelet transform	58
4.2 Wavelets with univariate generating kernels	61
4.3 Vertex-dynamic spectral graph wavelets	63
4.4 Numerical Implementation	66
4.5 Lack of Multiresolution Analysis	70
5 Spectral Graph Sparsification	79
5.1 Optimality of Spielman's Twice-Ramanujan Sparsification	80
6 Graph Conditioning and Optimization	88
6.1 Condition number and scalable frames	88
6.2 Minimizing condition number of graphs	92
6.3 Other methods of rescaling graphs	100
6.4 Interpretation of graph conditioning	105
Bibliography	110

List of Notations

\mathbb{R}	the field of real numbers
\mathbb{C}	the field of complex numbers
\mathbb{N}	the set of natural numbers $\{1, 2, 3, \dots\}$
V	the vertex set
E	the edge set
$\omega, \omega_{ij}, \omega(i, j)$	the edge weight function
$G, G(V, E)$	an unweighted graph
$G, G(V, E, \omega)$	an weighted graph
N	the cardinality of vertex set, $ V $
d_x	the degree of vertex x
$d(x, y)$	the shortest path between vertices x and y
$R(x, y)$	the effective resistance between vertices x and y
$\delta(i, j)$	the Kronecker delta function
L	the Laplacian matrix
D	the degree matrix
B	the incidence matrix
$\{\lambda_k\}_{k=0}^{N-1}$	the eigenvalues of Laplacian matrix
$\{\varphi_k\}_{k=0}^{N-1}$	the eigenvectors of Laplacian matrix
Φ	the eigenvector matrix
M_k	the k th graph modulation operator
T_i	the i th graph translation operator
T_g	the graph wavelet operator with generating kernel g
$\mathbb{1}_N$	the $N \times 1$ vector of all ones
$\mathbb{1}_S$	the indicator function on the set S
$\kappa, \kappa(L), \kappa(G)$	the condition number of the Laplacian matrix L for graph G
K_N	the complete graph on N vertices
P_N	the path graph on N vertices
C_N	the cycle graph on N vertices
SG, SG_n	the Sierpinski gasket and its n 'th graph approximation, respectively

Chapter 1

Introduction

Spectral graph theory has developed into a valuable tool for data analysis. In this age of big data, there is a need to efficiently analyze data with a network structure including social networks, molecular configurations, brain structure, and the world-wide web, to name a few. Spectral graph theory uses analytic properties of operators on these graph structures to unearth particular properties and glean insight into those spaces.

Harmonic analysis is concerned with the representation of functions and signals, be it through Fourier series, wavelets, or frames. Time-frequency analysis, dimensionality reduction, and image processing all are based on theory that is deeply founded in harmonic analysis and are all valuable tools in this age of plentiful digital data.

We shall bridge the world between spectral graph theory and harmonic analysis to develop tools and techniques to analyze functions and signals on graphs. The field is young and rapidly developing and this thesis is an exploration into the wild frontier that is harmonic analysis on graphs.

1.1 Linear Algebra

In computational and applied problems in mathematics and engineering, matrix analysis is often a crucial tool. The discrete and finite nature of graphs lead themselves to represent functions and operators on graphs by finite vectors and matrices, respectively.

We represent the space of $d \times m$ complex-valued (resp. real-valued) matrices as $\mathbb{C}^{d \times m}$ (resp. $\mathbb{R}^{d \times m}$). We denote the conjugate transpose of matrix A by A^* , i.e., $A_{ij}^* = \overline{A_{ji}}$. In the case that A is real-valued, then the conjugate transpose A^* agrees with the usual transpose A^\top . The square matrix $A \in \mathbb{C}^{d \times d}$ is Hermitian if $A = A^*$. In the case that A is real-valued, being Hermitian is equivalent to being symmetric. The Hermitian matrix $A \in \mathbb{C}^{d \times d}$ is positive semidefinite provided for any vector $x \in \mathbb{C}^d$ then the quantity $\langle Ax, x \rangle = x^* Ax$ is real-valued and nonnegative. We shall let $A \succeq 0$ to denote that matrix A is positive semidefinite and $A \succeq B$ to denote that $A - B$ is positive semidefinite.

There are three commonly used norms for matrices $A \in \mathbb{C}^{d \times m}$. The spectral norm is defined to be the largest singular value of A , denoted $\|A\|_2 = \sigma_{\max}(A)$; the operator norm is defined as $\|A\|_{\text{op}} = \sup_{x \neq 0} \|Ax\| / \|x\|$; and the Frobenius norm, or Hilbert-Schmidt norm, is the ℓ^2 -norm of all entries in A , i.e., $\|A\|_F = \sqrt{\sum_{i=1}^d \sum_{j=1}^m |a_{ij}|^2}$.

We let $\sigma(A) = \{\lambda_k\} \subset \mathbb{C}$ denote the spectrum of matrix A , that is the collection of scalars, known as eigenvalues, that solve $A\varphi = \lambda\varphi$ for $\varphi \in \mathbb{C}^d$ and $\varphi \neq 0$. In this case, we say that φ is the eigenvector associated with eigenvalue λ .

Theorem 1.1.1 (Spectral Theorem of Hermitian Matrices, [38]). *If $A \in \mathbb{R}^{d \times d}$ is Hermitian then all eigenvalues of A and the corresponding eigenvectors are real-valued. Furthermore, if A is positive semi-definite, then its eigenvalues are all non-negative.*

The Courant-Fischer theorem (also known as the min-max theorem) characterizes the eigenvalues of a Hermitian matrix, A , using the Rayleigh quotient of a non-zero vector x defined as

$$R_A(x) = \frac{x^*Ax}{x^*x}.$$

Theorem 1.1.2 (Courant-Fischer Theorem, [38]). *Let A be a Hermitian matrix with real eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$. Then,*

$$\lambda_k = \max_{\substack{S \subseteq \mathbb{R}^N \\ \dim(S) = N - k + 1}} \min_{\substack{x \in S \\ x \neq 0}} \frac{x^*Ax}{x^*x} = \min_{\substack{T \subseteq \mathbb{R}^N \\ \dim(T) = k}} \max_{\substack{x \in T \\ x \neq 0}} \frac{x^*Ax}{x^*x},$$

where the first maximum is taken over all subspaces, S , of dimension $N - k + 1$ and the last minimum is taken over all subspaces, T , of dimension k . In particular, we have

$$\lambda_N = \max_{x \in \mathbb{R}^N} \frac{x^*Ax}{x^*x} \quad \text{and} \quad \lambda_1 = \min_{x \in \mathbb{R}^N} \frac{x^*Ax}{x^*x}.$$

The Courant-Fischer theorem provides an iterative scheme that numerical solvers deploy in approximation of the spectrum of a Hermitian matrix.

Let A be a Hermitian matrix with real eigenvalues $\{\lambda_k\}_{k=1}^N$ and associated orthonormal eigenvectors $\{\varphi_k\}_{k=1}^N$. Then A can be written as a sum of rank-one matrices as

$$A = \sum_{k=1}^N \lambda_k \varphi_k \varphi_k^*.$$

If A is nonsingular then an inverse matrix A^{-1} exists. However, if A is singular, equivalently $0 \in \sigma(A)$, then we resort to a generalization of the notion of an inverse matrix known as the Moore-Penrose pseudoinverse of A , which we shall henceforth refer to as just the psuedoinverse of A .

Definition 1.1.3. The *pseudoinverse* of the singular Hermitian matrix A , denoted by A^+ , is defined as $A^+ = (A^*A)^{-1}A^*$ which acts as the inverse of A restricted to vectors projected onto the column space of A . Equivalently, the pseudoinverse can be expressed as

$$A^+ = \sum_{k:\lambda_k \neq 0} \frac{1}{\lambda_k} \varphi_k \varphi_k^*.$$

1.2 Graph Definitions

This section contains essential definitions and preliminary results on graphs that will be used throughout this document. The results presented here can all be found in [17, 60].

An unweighted graph is defined by the pair (V, E) where V denotes the set of vertices and E denotes the set of edges. When the vertex and edge set (V, E) are clear, we will simply denote the graph by G , otherwise when necessary, we will denote the graph by $G(V, E)$ to indicate the vertices and edges in consideration. Unless explicitly stated otherwise, we will consider only finite graphs, meaning the vertex set has finite cardinality. We reserve N to denote the cardinality of V . Each element in the edge set E is denoted as an ordered pair (x, y) where $x, y \in V$. If $(x, y) \in E$, we will often write $x \sim y$ to indicate that vertex x is connected to y . In

such case, we say that y is a neighbor of x .

A graph is *undirected* if the edge set is symmetric, that is $(x, y) \in E$ implies $(y, x) \in E$. For the entirety of this document, we will consider only undirected graphs. The graph is *simple* if there are no self-loops, that is, the edge set contains no edges of the form (x, x) . Additionally, we assume that graphs have only zero or one edge between any two pair of vertices, i.e., we do not allow multiple-edges between vertices; such graphs are known as multi-graphs and will not be considered in this document.

A weighted graph is defined by the triple $G(V, E, \omega)$, where $\omega : E \rightarrow [0, \infty)$ is known as the *weight function* that assigns a nonnegative weight to every edge in the set E . The weight of edge (x, y) is denoted $\omega(x, y)$ or ω_{xy} as shorthand. If $(x, y) \notin E$, we define $\omega_{xy} = 0$. When graphs are used as a model of a physical system, greater weights are often used to differentiate and emphasize strengthened connections, proximities, or similarities between points.

The *degree* of vertex $x \in V$ in an unweighted undirected graph equals the number of edges emanating from (equivalently, to) x and is denoted d_x . For an undirected weighted graph, the degree equals the sum of edge weights containing x , i.e.,

$$d_x = \sum_{y \in V: (x,y) \in E} \omega_{xy}.$$

These two definitions of degree are consistent. An unweighted graph $G(V, E)$ can simply be viewed as a weighted graph $G(V, E, \omega)$ where ω is identically 1 on E . In such case, $d_x = \sum_y \omega_{xy} = \sum_{y \sim x} 1$ equals precisely the number of edges containing

the vertex x . A graph is called *regular* if every vertex has the same degree; it is called k -regular when that degree equals $k \in \mathbb{N}$.

Consider the graph $G(V, E)$ and a subset of vertices $S \subseteq V$. The *subgraph* induced by S is the graph with vertex set S and edge set consisting of all edges in E with both endpoints in S .

We will consider functions on graphs that take on real (or complex) values on the vertices of the graph. Since $V = \{x_i\}_{i=1}^N$ is finite, it is often useful, especially when doing numerical computations, to represent $f : V \rightarrow \mathbb{R}$ as a vector of length N whose i 'th component equals $f(x_i)$.

A *path* (of length m), denoted p , is defined to be a sequence of adjacent edges, $p = \{(p_{j-1}, p_j)\}_{j=1}^m$. We say that the path p connects p_0 to p_m . A path is said to be simple if no edge is repeated in it. A graph is *connected* if for any two distinct vertices $x, y \in V$, there exists some path connecting x and y . The distance between any two vertices $x, y \in V$ is the length of the shortest path connecting x to y and is denoted $d(x, y)$. Let $S, T \subseteq V$ be two vertex subsets. The distance between sets S and T is defined as $\min\{d(x, y) : x \in S, y \in T\}$.

1.2.1 The Laplacian and other matrices associated with graphs

Given a finite unweighted graph $G(V, E)$ the adjacency, the *adjacency matrix* is the $N \times N$ matrix, A , defined by

$$A(i, j) = \begin{cases} 1, & \text{if } x_i \sim x_j \\ 0, & \text{otherwise.} \end{cases}$$

The analogue of the adjacency for a weighted graph, denoted W , is defined by $W(i, j) = \omega_{ij}$.

The *degree matrix* is the $N \times N$ diagonal matrix D whose entries equal the degrees d_{x_i} , i.e.,

$$D(i, j) = \begin{cases} d_{x_i}, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases}$$

The main differential operator that we shall study is L , the Laplacian (or Laplace's operator). For a function $f : \mathbb{R} \rightarrow \mathbb{R}$, the Laplacian is precisely the second derivative, for which we have the difference formula

$$f''(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \quad (1.1)$$

Suppose we were to discretize the real line, say, by the dyadic points, i.e., points that can be written as $k/2^n$ for $k \in \mathbb{Z}$, $n \in \mathbb{N}$, and form a graph. Each vertex in this graph would have an edge connecting it to its two closest neighbors. Then the second difference quotient formula (1.1) would become

$$f''(x) = \lim_{n \rightarrow \infty} \frac{f(x + \frac{1}{2^n}) - 2f(x) + f(x - \frac{1}{2^n})}{(\frac{1}{2^n})^2} \quad (1.2)$$

Notice that (1.2) is the sum of all the differences of $f(x)$ with f evaluated at all its neighbors (and then properly renormalized).

Motivated by this construction on the real line, we define the Laplacian in a graph setting. The pointwise formulation of the unweighted Laplacian applied to a function $f : V \rightarrow \mathbb{R}$ is defined as

$$Lf(x) = \sum_{y \sim x} f(x) - f(y). \quad (1.3)$$

This is precisely the same formulation given in [17, 67] (up to a sign). For a weighted graph, the formulation in (1.3) is generalized as $Lf(x) = \sum_{y \sim x} \omega_{xy}(f(x) - f(y))$.

For a finite graph, Laplace's operator can be represented as a matrix. As an abuse of notation, we will interchangeably use L to represent the Laplacian operator and also the Laplacian matrix. For every point $x_i \in V$, (1.3) gives a system of linear equations depending on function values at other points in V . Each of these linear equations can be represented in the rows of the Laplacian matrix yielding

$$L(i, j) = \begin{cases} d_{x_i} & \text{if } i = j \\ -1 & \text{if } x_i \sim x_j \\ 0 & \text{otherwise,} \end{cases} \quad (1.4)$$

or, equivalently, $L = D - A$. In the case that the graph G is weighted, then we have $L = D - W$.

Matrix L is called the unnormalized Laplacian to distinguish it from the normalized Laplacian, $\mathcal{L} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}AD^{-1/2}$, used in some of the literature on graphs, e.g., [17]. However, we shall work exclusively with the unnormalized Laplacian and shall henceforth just refer to it as the Laplacian.

For an undirected graph, let e_i denote the $N \times 1$ vector defined by $e_i(j) = \delta(i, j)$ and for every $\ell = 1, \dots, m = |E|$, let $v_\ell = e_i - e_j$, where (i, j) is the ℓ 'th edge in the edge set E . The *incidence matrix* of G , denoted B , is the $N \times m$ matrix $B = [v_1 v_2 \cdots v_m]$. The incidence matrix can be used to give yet another characterization of the Laplacian matrix, $L = BB^*$. This gives the Laplacian matrix

as a sum of rank-1 matrices, namely,

$$L = \sum_{\ell=1}^m v_{\ell} v_{\ell}^* \tag{1.5}$$

Observe that in the definition of the incidence matrix, the ordering of i and j in $v_{\ell} = e_i - e_j$, were not specified and hence the incidence matrix B is not unique. Furthermore, multiplying any of the columns of B by -1 does not affect (1.5).

1.2.2 Elementary properties of the Laplacian

By the Courant-Fischer theorem, Theorem 1.1.1, since L is a real symmetric matrix, it has real eigenvalues $\{\lambda_k\}_{k=0}^{N-1}$ with associated orthonormal eigenvectors $\{\varphi_k\}_{k=0}^{N-1}$. Furthermore, since $L = BB^*$ we immediately can state that the Laplacian is a positive semidefinite matrix and hence all of its eigenvalues are nonnegative. Let us define Φ to be the $N \times N$ orthogonal matrix whose $(k-1)$ th column is the vector φ_k . The spectrum of the Laplacian, $\sigma(L)$, is fixed but one's choice of eigenvectors $\{\varphi_k\}_{k=0}^{N-1}$ can vary and hence Φ is not unique. For instance, we can multiply any column of Φ by -1 since if φ is an eigenvector of L then so is $-\varphi$. Furthermore, if an eigenvalue λ has multiplicity $m > 1$, then any orthonormal basis for that m -dimensional eigenspace will form m linearly independent eigenvectors associated with λ . So while Φ is not unique, for the theory that follows, we assume that one has fixed an eigenbasis and hence the matrix Φ is assumed to be fixed.

The following is a cornerstone in characterizing the relationship between eigenvalues of a graph and connectedness properties of the graph.

Theorem 1.2.1 ([17]). *If the graph G is connected then $\lambda_0 = 0$ and $\lambda_i > 0$ for*

all $1 \leq k \leq N - 1$. In this case $\varphi_0 \equiv 1/\sqrt{N}$. More generally, if the graph G has m connected components, then $\lambda_0 = \lambda_1 = \dots = \lambda_{m-1} = 0$ and $\lambda_k > 0$ for all $k = m, \dots, N - 1$. The indicator function on each connected component (properly renormalized), forms an orthonormal eigenbasis for the m -dimensional eigenspace associated to eigenvalue 0.

We index the eigenvalues from 0 to $N - 1$ (rather than 1 to N) because the any graph will have at least one eigenvalue equal to zero and a connected graph will have λ_1 equal its first nonzero eigenvalue. This first nonzero eigenvalue is known as the *algebraic connectivity* of the graph, see [29], and is widely studied. The algebraic connectivity gets its name because it is a measure of how connected a graph is. Its corresponding eigenvector, φ_1 , is known as the *Fiedler vector* and will be discussed more in-depth in Chapter 3

The graph Laplacian has been used to generalize standard results from calculus and analysis on graph domains. In [41, 53, 56, 57, 67] and references therein, the authors use the graph Laplacian to generalize theory calculus and differential equations to graphs but only the specific case of graph approximations to fractals. The authors of [31, 32] derive the wave equation on more general graphs and use the graph Laplacian to derive Sobolev-type inequalities however their theory applies only for functions defined on the edge set E , rather than the vertex set V which is more commonly the case in the graph literature. In Chapter 2 we will use the graph Laplacian to mimic properties of the Laplacian in Euclidean space to define the graph analogues of time-frequency operators.

We conclude this subsection by defining the graph effective resistance.

Definition 1.2.2. For any two vertices, $i \neq j \in V$, the *effective resistance* between i and j is defined as

$$R(i, j) = (e_i - e_j)^\top L^+(e_i - e_j),$$

where e_i is the vector of all zeros except for a 1 in the i th position. When $i = j$, we define the effective resistance to be $R(i, i) = 0$.

See [25] for a survey and introduction to effective resistance on graphs. The effective resistance gets its name because it is derived from the physical interpretation of the graph as an electrical network. Interpret the vertices of the graphs as nodes and each edge as a resistor with conductance ω_{ij} . The effective resistance between two nodes is the potential difference between those nodes when a unit current is injected at one node and extracted at the other (this is the role of $e_i - e_j$).

Strichartz gives an interesting equivalent definition for effective resistance in [67, Section 1.6] given as

$$R(i, j)^{-1} = \min_{f: V \rightarrow \mathbb{R}} \{\langle Lf, f \rangle : f(i) = 1 \text{ and } f(j) = 0\}.$$

The effective resistance is studied because it can be shown, see [67], that the effective resistance is a metric. The effective resistance is often times a more insightful metric than shortest path metric or the Euclidean metric in which the graph is embedded. It has been shown in [15, 24] that the effective resistance is equivalent to the commute time between vertices in a random walk on graphs.

1.3 Data graphs and common graphs

This section is devoted to introducing common graphs in the spectral graph theory literature and their properties. We also introduce some graphs arising from real-world data sets.

The *path graph* on N vertices is denoted P_N . The edge set contains the $N - 1$ edges given by $E = \{(x_i, x_{i+1})\}_{i=1}^{N-1}$. It is called the path graph because the entire graph is a path connecting x_1 to x_N . Figure 1.1 shows an illustration of P_6 , the path graph on 6 vertices.



Figure 1.1: The path graph P_6 on six vertices.

The *cycle graph* on N vertices is denoted C_N . The graph is identical to the path graph, P_N , with the added edge (x_1, x_N) connecting the two endpoints of the graph, thus forming a cycle. Figure 1.2 shows an illustration of the cycle graph C_6 on six vertices.

The *complete graph* on N vertices is denoted K_N . The complete graph contains all possible edges allowed in a simple graph, i.e., $E = \{(x_i, x_j)\}_{i \neq j}$. As such, one can compute that the complete graph contains exactly $|E| = N(N - 1)/2$ edges, which is an upper bound of the number of edges for any simple graph. Figure 1.3 shows

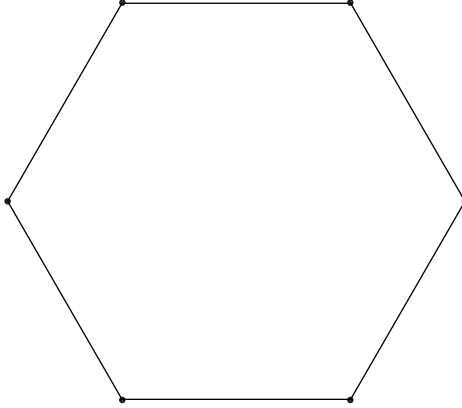


Figure 1.2: The cycle graph C_6 on six vertices.

an illustration of the complete graph on 6 and 25 vertices. The complete graph is significant because it is the most connected a graph can be; the distance between any two vertices is one. The Laplacian for the complete graph K_N has spectrum $\lambda_0 = 0$ and $\lambda_1 = \dots = \lambda_{N-1} = N$. It is the only graph for which $\lambda_1 = \lambda_{N-1}$; that is, the complete graph has the largest possible algebraic connectivity on N vertices.

The *Sierpinski gasket*, SG, is a post-critically finite fractal, which is an example of an infinite graph. If q_0, q_1, q_2 represent the three vertices of a triangle, then the Sierpinski gasket is the unique nonempty compact solution to $\text{SG} = \cup_{i=0}^2 F_i(\text{SG})$, where $F_i x := 1/2(x - q_i) + q_i$ for $i = 0, 1, 2$ are contraction mappings to each of the three corners of the triangle. As is done in [41, 53, 67], we consider finite graph approximations to the fractal. The m 'th graph approximation to the Sierpinski gasket, denoted SG_m , is the image of the triangle q_0, q_1, q_2 under all possible compositions of $\{F_i\}_{i=0}^2$ of length m . Figure 1.4 shows some graph approximations to the full

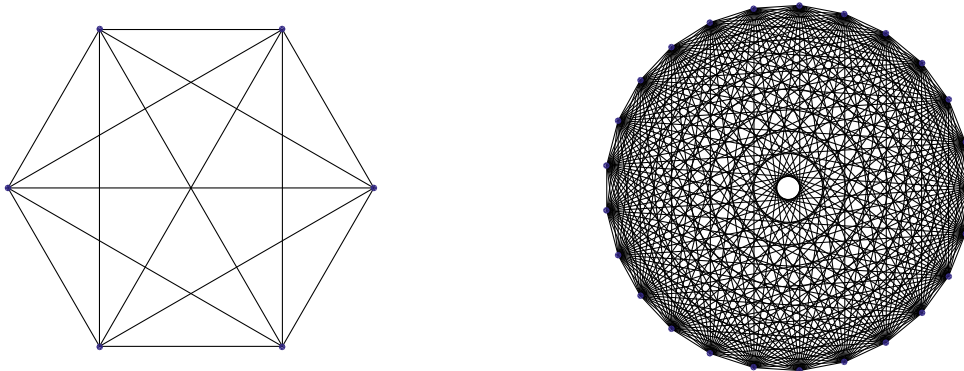


Figure 1.3: Left: The complete graph K_6 on six vertices. Right:
The complete graph K_{25} on 25 vertices.

Sierpinski gasket.

Finally, we shall demonstrate numerical implementations of theory developed on a real-life dataset known as the *Minnesota road network*. The Minnesota road network is available for download as part of the Sparse Matrix Collection hosted by the University of Florida [23]. The network consists of 2640 nodes and 3302 edges. Edges represent major roads in the US state of Minnesota and vertices represent intersections of those roads. Figure 1.5 shows the Minnesota road network in its given coordinates.

1.4 Outline of results

In Chapter 2 we present the newly-developed theory of Fourier analysis on graphs introduced in [59]. We primarily focus on the graph translation operator in Section 2.4 since it has substantial differences to the classical Euclidean analogue of

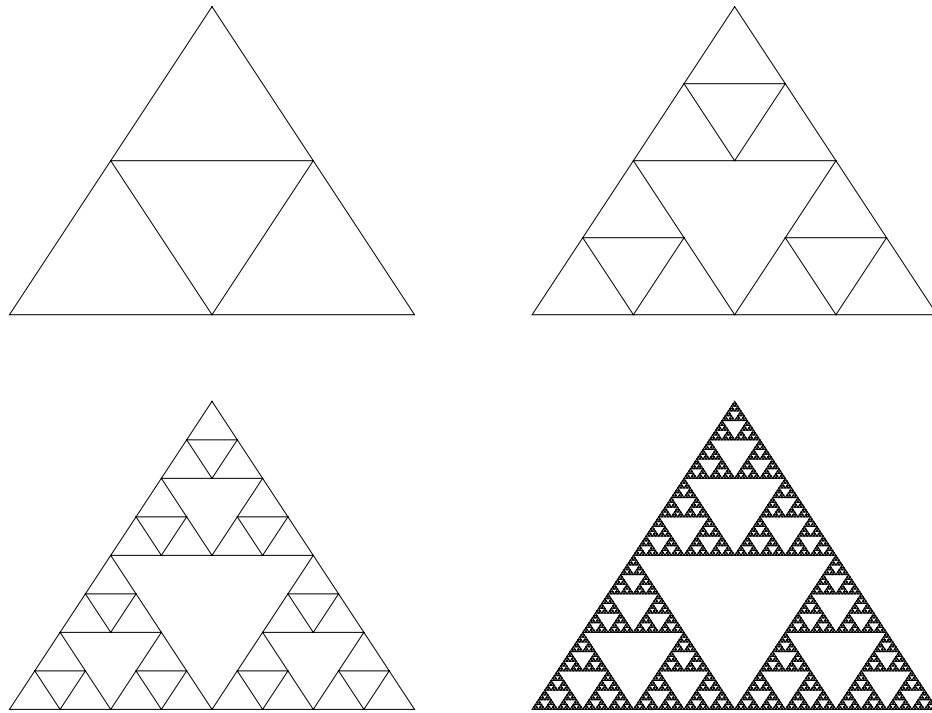


Figure 1.4: The graph approximations SG_1 , SG_2 , SG_3 , to the full Sierpinski gasket shown on the bottom right.

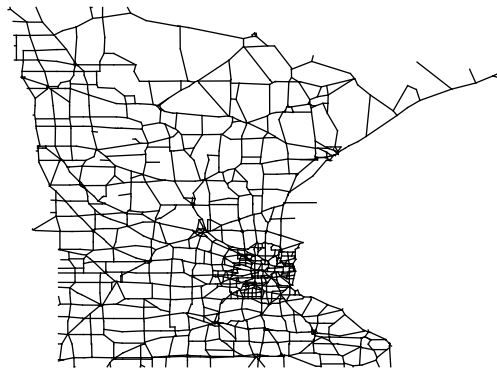


Figure 1.5: The Minnesota Road network

translation. We prove when the graph translation operator acts as a semigroup and completely characterize conditions in which the operator is invertible and derive its inverse.

In Chapter 3 we examine characteristic sets of eigenvectors of the Laplacian because it is directly related to the theory of translation developed in Section 2.4. In particular, we focus on the support of the Fiedler vector of the graph. We prove in Section 3.1 that planar graphs cannot have large neighborhoods of vertices on which the Fiedler vector vanishes. We then introduce a family of (non-planar) graphs, called barren graphs, that have arbitrarily large neighborhoods on which the Fiedler vector does vanish in Section 3.1.1. In Section 3.1.2 we prove results about the algebraic connectivity and Fiedler vector of a graph formed by adding multiple graphs.

Chapter 4 presents the spectral graph wavelets developed in [35]. We modify the existing theory in Section 4.3 to create graph wavelets with a bivariate generating kernel which gives the user vertex-dependent control of the size and redundancy of the wavelet frame. Finally, we prove in Section 4.5 that unlike classical wavelets, the spectral graph wavelets do not admit a multiresolution analysis.

Chapter 5 explores the concept of graph sparsification pioneered by Daniel Spielman and his collaborators which has many applications in fast computation on large networks. We briefly introduce Spielman's twice-Ramanujan sparsification results from [4]. Spielman conjectures the optimal sparsification bounds of any graph. We prove that his algorithm cannot possibly exceed this conjectured bound.

Chapter 6 introduces a new concept of graph conditioning. Related to scal-

able frames and Spielman's sparsification results, the goal is to rescale edges of a graph to make its Laplacian act as an approximate identity. There are several ways to approach this problem: via condition number (Section 6.2), spectral gap, and Frobenius distance to identity (Section 6.3). We present numerical experiments and analysis for each of these proposed methods.

Chapter 2

Fourier Analysis on Graphs

Recently in [59] Vandergheynst and his collaborators have presented a way to use the eigenfunctions of the graph Laplacian to define a Fourier transform of a function defined on a graph. They then take this definition along with classical properties in time-frequency analysis to define graph analogues of operators needed in analysis, namely, convolution, modulation, and translation. In this chapter, we analyze and expand on properties of the translation operator that they have introduced. In particular, we characterize in Theorem 2.4.8 exactly when the translation operator is invertible.

2.1 The Graph Fourier Transform

The classical Fourier transform on the real line is the expansion of a function f in terms of the eigenvalues of the eigenfunctions of the Laplace operator, i.e. $\hat{f}(\xi) = \langle f, e^{2\pi i \xi \cdot t} \rangle$. Analogously, we define the *graph Fourier transform*, \hat{f} , of a functions $f : V \rightarrow \mathbb{C}$ as the expansion of f in terms of the eigenfunctions of the graph Laplacian.

Definition 2.1.1. Given the graph, G , and its Laplacian, L , with spectrum $\sigma(L) = \{\lambda_k\}_{k=0}^{N-1}$ and eigenvectors $\{\varphi_k\}_{k=0}^{N-1}$, the *graph Fourier transform* of $f : V \rightarrow \mathbb{C}$ is by

$$\hat{f}(\lambda_k) = \langle f, \varphi_k \rangle = \sum_{n=1}^N f(n) \varphi_k^*(n). \quad (2.1)$$

Notice that the graph Fourier transform is only defined on values of $\sigma(L)$. There is a serious problem when λ_k has multiplicities greater than one because then \hat{f} will not be a well-defined function. Rather, one should interpret the notation $\hat{f}(\lambda_k)$ to designate the inner product of f with the k 'th eigenfunction of L . However to emphasize that the Fourier transform is defined in the spectral domain, we shall abuse the notation as defined here.

The *graph inverse Fourier transform* is then given by

$$f(n) = \sum_{l=0}^{N-1} \hat{f}(\lambda_l) \varphi_l(n). \quad (2.2)$$

If we consider the function f and \hat{f} as $N \times 1$ vectors, then (2.1) and (2.2) become

$$\hat{f} = \Phi^* f \quad \text{and} \quad f = \Phi \hat{f}.$$

Since, Φ is a unitary matrix, the derivation of (2.2) is a direct result of (2.1). Futhermore, (2.2) implies that every function equals the analogue of what would be its Fourier series using the definitions provided here.

Proposition 2.1.2 (Parseval's Identity). *For any $f, g : V \rightarrow \mathbb{R}$, then Parseval's relation holds. That is, $\langle f, g \rangle = \langle \hat{f}, \hat{g} \rangle$. Moreover, this implies that*

$$\|f\|_{\ell^2}^2 = \sum_{n=1}^N |f(n)|^2 = \sum_{l=0}^{N-1} |\hat{f}(\lambda_l)|^2 = \|\hat{f}\|_{\ell^2}^2.$$

Proof. Since the matrix Φ is unitary, then we can compute directly

$$\langle \hat{f}, \hat{g} \rangle = \hat{f}^* \hat{g} = (\Phi^* f)^* \Phi^* g = f^* \Phi \Phi^* g = f^* g = \langle f, g \rangle.$$

□

2.2 Graph Convolution

The convolution of two signals $f, g \in L^2(\mathbb{R})$ is defined as

$$f * g(t) = \int_{\mathbb{R}} f(u)g(t - u) du.$$

However, we have not yet established a clear analogue of translation in the graph setting. Instead, we exploit the following property of the convolution: $\widehat{(f * g)}(\xi) = \hat{f}(\xi)\hat{g}(\xi)$, see [6]. Then by taking the inverse graph Fourier transform, (2.2), we can define convolution in the graph domain. For $f, g : V \rightarrow \mathbb{R}$, we define the *graph convolution* of f and g as

$$f * g(n) = \sum_{l=0}^{N-1} \hat{f}(\lambda_l)\hat{g}(\lambda_l)\varphi_l(n). \quad (2.3)$$

Many of the classical time-frequency properties of the convolution hold for the graph convolution.

Proposition 2.2.1 ([59, Proposition 1]). *For $\alpha \in \mathbb{R}$, and $f, g, h : V \rightarrow \mathbb{R}$ then the graph convolution defined in (2.3) satisfies the following properties:*

1. $\widehat{f * g} = \hat{f}\hat{g}$.
2. $\alpha(f * g) = (\alpha f) * g = f * (\alpha g)$.

3. *Commutativity:* $f * g = g * f$.

4. *Distributivity:* $f * (g + h) = f * g + f * h$.

5. *Associativity:* $(f * g) * h = f * (g * h)$.

The proofs of these properties are straightforward and some follow immediately from Definition (2.3).

If we want to express the graph convolution as a vector then (2.3) is equivalent to $(\hat{f} \odot \hat{g})\Phi^*$ where \odot is the pointwise multiplication operator (written as $.*$ in MATLAB). In MATLAB, (recalling that $\hat{f} = \Phi^* f$) one would express the vector $f * g$ as

$$\Phi * ((\Phi' * f) .* (\Phi' * g))$$

or equivalently

$$\Phi * (\text{diag}(\Phi' * f) * \Phi' * g) \text{ or } \Phi * (\text{diag}(\Phi' * g) * \Phi' * f)$$

where the operation $\text{diag}(a)$ takes a vector, a , and returns a square matrix with vector a on the diagonal and zero elsewhere.

Example 2.2.2. In the classical Euclidean setting, convolution with the Dirac delta function acts as the identity, i.e.,

$$f(x) = (f * \delta_0)(x) = \int_{\mathbb{R}} f(t - x) \delta_0(t) dt,$$

where

$$\delta_c(x) = \begin{cases} 1, & \text{if } x = c \\ 0, & \text{if } x \neq c. \end{cases}$$

The graph convolution also has a function that acts as the identity under the graph convolution. Consider the function $g_0 : V \rightarrow \mathbb{R}$ by first defining it in the spectral domain by $\hat{g}_0(\lambda_\ell) = 1$ for all $l = 0, \dots, N - 1$. Then the values of g_0 can be obtained by taking the inverse Fourier transform (2.2) giving $g_0(n) = \sum_{l=0}^{N-1} \varphi_l(n)$ (in vector notation, $g_0 = \Phi * \mathbb{1}_{N \times 1}$, where $\mathbb{1}_{N \times 1}$ is the $N \times 1$ vector of all ones). Then as a result of Proposition 2.2.1, we have

$$\begin{aligned} f(n) &= \sum_{l=0}^{N-1} \hat{f}(\lambda_\ell) \varphi_l(n) = \sum_{l=0}^{N-1} \hat{f}(\lambda_\ell) \hat{g}_0(\lambda_\ell) \varphi_l(n) \\ &= f * g_0(n). \end{aligned} \tag{2.4}$$

In other words, $f * g_0 = f$, so convolution with the function g_0 is the identity.

2.3 Graph Modulation

Motivated by the fact that in Euclidean space, modulation of a function is multiplication of a Laplacian eigenfunction, we define for any $k = 0, 1, \dots, N - 1$ the *graph modulation operator* $M_k : \mathbb{R}^N \rightarrow \mathbb{R}^N$ as

$$(M_k f)(n) = \sqrt{N} f(n) \varphi_k(n). \tag{2.5}$$

Notice that since $\varphi_0 \equiv \frac{1}{\sqrt{N}}$ then M_0 is the identity operator.

An important remark is that in the classical case, modulation in the time domain represents translation in the frequency domain, i.e., $\widehat{M_\xi f}(\omega) = \hat{f}(\omega - \xi)$, see [6]. The graph modulation does not exhibit this property due to the discrete nature of the spectral domain. However, it is worthy to notice the special case if

$\hat{g}(\lambda_\ell) = \delta_0(\lambda_\ell)$, i.e., g is a constant function, then

$$\widehat{M_k g}(\lambda_\ell) = \sum_{n=1}^N \varphi_\ell^*(n)(M_k g)(n) = \sum_{n=1}^N \varphi_\ell^*(n)\sqrt{N}\varphi_k(n)\frac{1}{\sqrt{N}} = \delta_\ell(k).$$

That is, modulation M_k sends the constant function to the k 'th eigenvector, which, in the spectral domain, translates $\hat{g} = \delta_0$ to $\widehat{M_k g} = \delta_k$.

We can express the operator M_k as a diagonal matrix

$$M_k = \begin{pmatrix} \varphi_k(1) & & 0 \\ & \ddots & \\ 0 & & \varphi_k(N) \end{pmatrix}$$

and in the language of MATLAB we have

$$\mathbf{Mk} = \mathbf{diag}(\Phi(:, \mathbf{k})).$$

2.4 Graph Translation

For signal $f \in L^2(\mathbb{R})$, the translation operator, T_u , which translates f by vector u , can be thought of as a convolution with δ_u . Then in \mathbb{R} , by exploiting (2.3) we have

$$(T_u f)(t) = f(t - u) = (f * \delta_u)(t) = \int_{\mathbb{R}} \hat{f}(k)\hat{\delta}_u(k)\varphi_k(t) dk = \int_{\mathbb{R}} \hat{f}(k)\varphi_k^*(u)\varphi_k(t) dk$$

since $\hat{\delta}_u(k) = \int_{\mathbb{R}} \delta_u(x)\varphi_k^*(x) dx = \varphi_k(u)$.

Motivated by this example, for any $f : V \rightarrow \mathbb{R}$ we can define the *graph translation operator*, $T_i : \mathbb{R}^N \rightarrow \mathbb{R}^N$ via the graph convolution of the Dirac delta centered at the i 'th vertex:

$$(T_i f)(n) = \sqrt{N}(f * \delta_i)(n) = \sqrt{N} \sum_{k=0}^{N-1} \hat{f}(\lambda_k) \varphi_k^*(i) \varphi_k(n). \quad (2.6)$$

We can express $T_i f$ in matrix notation as follows:

$$\begin{aligned} T_i f &= \sqrt{N} \begin{pmatrix} \varphi_0^*(i) \varphi_0(1) & \varphi_1^*(i) \varphi_1(1) & \cdots & \varphi_{N-1}^*(i) \varphi_{N-1}(1) \\ \vdots & \vdots & \cdots & \vdots \\ \varphi_0^*(i) \varphi_0(N) & \varphi_1^*(i) \varphi_1(N) & \cdots & \varphi_{N-1}^*(i) \varphi_{N-1}(N) \end{pmatrix} \begin{pmatrix} \hat{f}(\lambda_0) \\ \vdots \\ \hat{f}(\lambda_{N-1}) \end{pmatrix} \\ &= \sqrt{N} \Phi \square (\varphi_0^*(i), \dots, \varphi_{N-1}^*(i)) \Phi^* f \end{aligned} \quad (2.7)$$

where we introduce the binary operation \square to signify element-wise multiplication.

For example, if

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad b = (b_1, b_2)$$

then $A \square b$ multiplies the row vector b component-wise with each row of A ; that is

$$A \square b = \begin{pmatrix} a_{11} b_1 & a_{12} b_2 \\ a_{21} b_1 & a_{22} b_2 \end{pmatrix}.$$

Similarly if b' was instead a column vector, i.e.

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad b' = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix},$$

then $A \square b'$ multiplies the column vector b' component-wise with each column of A ;

that is

$$A \square b' = \begin{pmatrix} a_{11} b_1 & a_{12} b_1 \\ a_{21} b_2 & a_{22} b_2 \end{pmatrix}.$$

The MATLAB command for $A \square b$ is `bsxfun(@times,A,b)`. Therefore for any

$i = 1, \dots, N$ the MATLAB command for the translation operator as

$$T_i = \sqrt{N} \text{bsxfun}(@\text{times}, \Phi, \Phi(i, :)) * \Phi'.$$

The following results follow from the properties of the graph convolution from Proposition 2.2.1:

Proposition 2.4.1. *For any $f, g : V \rightarrow \mathbb{C}$ and for any $i, j \in \{1, 2, \dots, N\}$ then*

1. $T_i(f * g) = (T_i f) * g = f * (T_i g)$.
2. $T_i T_j f = T_j T_i f$.

Proof. The proof of Property 1 follows from the definition of convolution (2.3). To prove Property 2 observe that

$$\begin{aligned}
T_i T_j f(n) &= \sqrt{N} \sum_{k=0}^{N-1} (\widehat{T_j f})(\lambda_k) \varphi_k^*(i) \varphi_k(n) \\
&= N \sum_{k=0}^{N-1} \sum_{m=1}^N (T_j f)(m) \varphi_k^*(m) \varphi_k^*(i) \varphi_k(n) \\
&= N \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} \sum_{m=1}^N \hat{f}(\lambda_\ell) \varphi_\ell^*(j) \varphi_k^*(i) \varphi_k(n) \varphi_\ell(m) \varphi_k^*(m) \\
&= N \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} \hat{f}(\lambda_\ell) \varphi_\ell^*(j) \varphi_k^*(i) \varphi_k(n) \delta_k(\ell) \\
&= N \sum_{k=0}^{N-1} \hat{f}(\lambda_k) \varphi_k^*(j) \varphi_k^*(i) \varphi_k(n) \\
&= N \sum_{k=0}^{N-1} (\widehat{f * \delta_i})(\lambda_k) \varphi_k^*(j) \varphi_k(n) = \sqrt{N} T_j (f * \delta_i)(n) = T_j T_i f(n).
\end{aligned}$$

□

The following result shows a reflexivity between the translation operator and the vertex at which it is evaluated.

Corollary 2.4.2. For any $i, n \in \{1, \dots, N\}$ and for any function $g : V \rightarrow \mathbb{C}$ we have

$$T_i g(n) = \overline{T_n \bar{g}(i)}.$$

Proof. By definition $(T_i g)(n) = \sqrt{N} \sum_{k=0}^{N-1} \hat{g}(\lambda_k) \varphi_k^*(i) \varphi_k(n)$. Hence,

$$\begin{aligned} \overline{(T_i g)(n)} &= \sqrt{N} \sum_{k=0}^{N-1} \overline{\hat{g}(\lambda_k) \varphi_k^*(i) \varphi_k(n)} \\ &= \sqrt{N} \sum_{k=0}^{N-1} \hat{g}(\lambda_k) \varphi_k(i) \varphi_k^*(n) = T_n \bar{g}(i), \end{aligned}$$

which proves the corollary. □

Notice that, in particular, Corollary 2.4.2 asserts that if the eigenvectors $\{\varphi_\ell\}_{\ell=0}^{N-1}$ are entirely real-valued, then

$$T_i g(n) = T_n g(i).$$

Combining this with Proposition 2.4.1 gives the following result.

Theorem 2.4.3. Assume G is a graph with real-valued eigenvectors $\{\varphi_k\}_{k=0}^{N-1}$. Let α be a multiindex, i.e. $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_K)$ where $\alpha_j \in \{1, \dots, N\}$ for $1 \leq j \leq K$ and let $\alpha_0 \in \{1, \dots, N\}$. We let T_α denote the composition $T_{\alpha_K} \circ \dots \circ T_{\alpha_2} \circ T_{\alpha_1}$. Then for any $f : V \rightarrow \mathbb{R}$, we have $T_\alpha f(\alpha_0) = T_\beta f(\beta_0)$ where $\beta = (\beta_1, \dots, \beta_K)$ and $(\beta_0, \beta_1, \beta_2, \dots, \beta_K)$ is any permutation of $(\alpha_0, \alpha_1, \dots, \alpha_K)$.

Proof. There exists a bijection between the collection of all possible $T_\alpha f(\alpha_0)$ for $|\alpha| = K$, $1 \leq \alpha_0 \leq N$, and the space of $(K+1)$ -tuples with values in $\{1, \dots, N\}$. That is, the map that sends $T_\alpha f(\alpha_0)$ to $(\alpha_0, \alpha_1, \dots, \alpha_K)$ is a bijection. This enables us to define an equivalence relation on the space $\{1, \dots, N\}^{K+1}$. We write $(a_0, \dots, a_K) \cong (b_0, \dots, b_K)$ if and only if $T_{a_K} \circ \dots \circ T_{a_1} f(a_0) = T_{b_K} \circ \dots \circ T_{b_1} f(b_0)$.

By Corollary 2.4.2, $(a_0, a_1, \dots, a_K) \cong \sigma_1(a_0, a_1, \dots, a_K) = (a_1, a_0, \dots, a_K)$, i.e., σ_1 is the permutation $(1, 2)$. In general, we write σ_i to denote the permutation $(i, i + 1)$.

By Proposition 2.4.1 b, $(a_0, a_1, \dots, a_K) \cong \sigma_i(a_0, a_1, \dots, a_K)$ for any $i = 2, 3, \dots, K - 1$.

We now have that any permutation σ_i for $i = 1, \dots, K - 1$ preserves equivalency. This collection of $K - 1$ transpositions allow for any permutation, which proves the corollary. \square

Proposition 2.4.1 and its corollaries conclude that the translation operator is distributive with the convolution and that the translation operators commute among themselves. However, the niceties end here; other properties of translation on the real line do not carry over to the graph setting. For example, we do not have the collection of graph translation operators forming a group, i.e., $T_i T_j \neq T_{i+j}$. In fact, we cannot even assert that the translation operators form a semigroup, i.e. $T_i T_j = T_{i \bullet j}$ for some semigroup operator $\bullet : \{1, \dots, N\} \times \{1, \dots, N\} \rightarrow \{1, \dots, N\}$. The following theorem characterizes graphs which do exhibit a semigroup structure of the translation operators.

Theorem 2.4.4. *Consider the graph, $G(V, E)$, with real-valued (resp. complex-valued) eigenvector matrix $\Phi = [\varphi_0 \cdots \varphi_{N-1}]$. Graph translation on G is a semigroup, i.e. $T_i T_j = T_{i \bullet j}$ for some semigroup operator $\bullet : \{1, \dots, N\} \times \{1, \dots, N\} \rightarrow \{1, \dots, N\}$, only if $\Phi = (1/\sqrt{N})H$, where H is a real-valued (resp. complex-valued) Hadamard matrix.*

Proof. *i.* We first show that graph translation on G is a semigroup, i.e. $T_i T_j = T_{i \bullet j}$

for some semigroup operator $\bullet : \{1, \dots, N\} \times \{1, \dots, N\} \rightarrow \{1, \dots, N\}$, if and only if $\sqrt{N}\varphi_k(i)\varphi_k(j) = \varphi_k(i \bullet j)$ for all $l = 0, \dots, N - 1$. By repeating the calculations in the proof of Proposition 2.4.1, we have

$$T_i T_j f(n) = N \sum_{k=0}^{N-1} \hat{f}(\lambda_k) \varphi_k^*(j) \varphi_k^*(i) \varphi_k(n)$$

and by definition

$$T_\ell f(n) = \sqrt{N} \sum_{k=0}^{N-1} \hat{f}(\lambda_k) \varphi_k^*(\ell) \varphi_k(n).$$

Therefore, $T_i T_j f = T_{i \bullet j} f$ will hold for any function $f : V \rightarrow \mathbb{R}$ if and only if $\sqrt{N}\varphi_k(i)\varphi_k(j) = \varphi_k(i \bullet j)$ for every $k \in \{0, \dots, N - 1\}$.

ii. We show next that $\sqrt{N}\varphi_k(i)\varphi_k(j) = \varphi_k(i \bullet j)$ for all $k = 0, \dots, N - 1$ only if the eigenvectors are constant amplitude, namely $1/\sqrt{N}$ by the orthonormality of the eigenvectors. Assume $\sqrt{N}\varphi_k(i)\varphi_k(j) = \varphi_k(i \bullet j)$, which, in particular, implies $\sqrt{N}\varphi_k(i)\varphi_k(i) = \sqrt{N}\varphi_k(i)^2 = \varphi_k(i \bullet i)$.

Suppose that $|\varphi_k(a_1)| < 1/\sqrt{N}$ for some $a_1 \in \{1, \dots, N\}$ and for some $k \in \{0, \dots, N - 1\}$. Then $\sqrt{N}\varphi_k(a_1)^2 < |\varphi_k(a_1)|$ and so $a_1 \bullet a_1 = a_2$ for some $a_2 \in \{1, \dots, N\} \setminus \{a_1\}$. Then since $|\varphi_k(a_2)| = \sqrt{N}\varphi_k(a_1)^2 < |\varphi_k(a_1)| < 1/\sqrt{N}$ we can repeat the same argument to assert $a_2 \bullet a_2 = a_3$ for some $a_3 \in \{1, \dots, N\} \setminus \{a_1, a_2\}$. This procedure can be repeated producing an infinite number of unique indices $\{a_i\}$ on a graph, G , which contradicts the graph having only $N < \infty$ nodes. A similar argument gives a contradiction if $|\varphi_k(i)| > 1/\sqrt{N}$ for any l, i . Therefore, the graph translation operators form a semigroup only if $|\varphi_k(n)| = 1/\sqrt{N}$ for all $k = 0, 1, \dots, N - 1$ and $n = 1, \dots, N$. Since Φ is an orthogonal matrix, i.e. $\Phi\Phi^* = \Phi^*\Phi = I$, then $\Phi = (1/\sqrt{N})H$, where H is a Hadamard matrix. \square

Remark 2.4.5. *a).* The order of a Hadamard matrix must be 1, 2, or a multiple of 4. Hadamard’s conjecture proposes that there exists a Hadamard matrix of order equal to every multiple of 4 and remains an open problem currently. See [36] for other properties and constructions of Hadamard matrices.

b). Rearranging rows and columns of a Hadamard matrix does not change its orthogonality nor its constant amplitude properties. Therefore, without loss of generality, we may assume that the first column of H is the vector of all ones to correspond with φ_0 equaling the constant vector with entries $1/\sqrt{N}$.

c). If we relax the constraint that Φ must be real-valued, we can obtain graphs with constant-amplitude eigenfunctions that allow the translation operators to form a (semi)group. For the cycle graph on N nodes, C_N , one can choose Φ equal to the discrete Fourier transform (DFT) matrix, where $\Phi_{nm} = e^{-2\pi i(n-1)(m-1)/N}$. Under this construction, we have $T_i T_j = T_{i+j \pmod{N}}$.

d). It is shown in [3, Theorem 5] that if $\Phi = (1/\sqrt{N})H$ for Hadamard H , then the spectrum of the Laplacian, $\sigma(L)$, must consist entirely of even integers. The authors of [27] explore graphs with integer spectrum but do not address the case of a spectrum of only even integers.

e). The converse to Theorem 2.4.4 does not necessarily hold. That is, if the eigenvector matrix $\Phi = 1/\sqrt{N}H$, for a renormalized Hadamard matrix H , then the translation operators on G need not form a semigroup. For example, consider the

real Hadamard matrix, H , of order 12 given by

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \end{bmatrix}.$$

Then the second and third columns multiplied componentwise equals the vector

$$[1, -1, 1, -1, -1, 1, 1, -1, 1, 1, -1, -1]^T,$$

which does not equal any of the columns of H .

What kinds of graphs have a Hadamard eigenvector matrix? The authors of [3] prove that if N is a multiple of 4 for which a Hadamard matrix exists, then the complete graph on N vertices, K_N , is one such graph.

Theorem 2.4.6 ([3, Observation 1]). *Suppose H is a real Hadamard matrix of order $N \in 4\mathbb{Z}$. Then $1/\sqrt{N}H$ is an eigenvector matrix for the complete graph of N*

vertices, K_N .

Proof. It is a standard result (see [17]) that the complete graph has Laplacian matrix $L = NI - J$, where J is the matrix with all entries 1, with eigenvalues $\lambda_0 = 0$ and $\lambda_i = N$ for $i = 1, \dots, N - 1$. Let D be the diagonal matrix $D(i, i) = \lambda_{i-1}$.

We can write the $N \times N$ matrix H as

$$H = \left[\mathbb{1} \mid \tilde{H} \right]$$

where $\mathbb{1}$ is the $N \times 1$ vector of all ones and \tilde{H} is the remaining $N \times (N - 1)$ matrix.

If the $N \times N$ matrix, $A = [a|B]$ where a is an $N \times 1$ vector, then it is simple to verify that

$$AA^\top = aa^\top + BB^\top.$$

This identity gives

$$NI = HH^\top = \mathbb{1}\mathbb{1}^\top + \tilde{H}\tilde{H}^\top,$$

where the first equality holds from the property of H being Hadamard. Additionally, one can compute $HDH^\top = n\tilde{H}\tilde{H}^\top$.

Thus we have

$$\left(\frac{1}{\sqrt{N}} H \right) D \left(\frac{1}{\sqrt{N}} H \right)^{-1} = \left(\frac{1}{N} \right) HDH^\top = NI - J = L,$$

which completes the proof. □

Also unlike in the classical case in \mathbb{R}^d , translation is not an isometric operation.

That is $\|T_i f\|_2 \neq \|f\|_2$. However, we do have the following estimates:

Lemma 2.4.7 ([59, Lemma 1]). *For any $f : V \rightarrow \mathbb{R}$,*

$$|\hat{f}(0)| \leq \|T_i f\|_2 \leq \sqrt{N} \max_{k \in \{0,1,\dots,N-1\}} |\varphi_k(i)| \|f\|_2 \leq \sqrt{N} \max_{k \in \{0,1,\dots,N-1\}} \|\varphi_k\|_\infty \|f\|_2 \quad (2.8)$$

Proof. By definition

$$\begin{aligned} \|T_i f\|_2^2 &= N \sum_{n=1}^N \left(\sum_{k=0}^{N-1} \hat{f}(\lambda_k) \varphi_k^*(i) \varphi_k(n) \right)^2 \\ &= N \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} \hat{f}(\lambda_k) \hat{f}(\lambda_{k'}) \varphi_k^*(i) \varphi_{k'}^*(i) \sum_{n=1}^N \varphi_k(n) \varphi_{k'}(n) \\ &= N \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} \hat{f}(\lambda_k) \hat{f}(\lambda_{k'}) \varphi_k^*(i) \varphi_{k'}^*(i) \sum_{n=1}^N \delta(k, k') \end{aligned}$$

where the last equality follows from the fact that $\{\varphi_k\}_{k=0}^{N-1}$ is an orthonormal basis and we can a priori choose the eigenvectors to all be real-valued. Then we can further simplify to obtain

$$\|T_i f\|_2^2 = N \sum_{k=0}^{N-1} |\hat{f}(\lambda_k)|^2 |\varphi_k^*(i)|^2 \leq N \max_{l=0,\dots,N-1} |\varphi_k(i)|^2 \|f\|_2^2 \leq N \max_{l \in \{0,1,\dots,N-1\}} \|\varphi_k\|_\infty^2 \|f\|_2^2.$$

The lower bound comes from the fact that $\varphi_0(n) = 1/\sqrt{N}$ for all n and so

$$\|T_i f\|_2^2 = N \sum_{k=0}^{N-1} |\hat{f}(\lambda_k)|^2 |\varphi_k^*(i)|^2 \geq N |\hat{f}(\lambda_0)|^2 |\varphi_0(i)|^2 = N |\hat{f}(0)|^2 \left| \frac{1}{\sqrt{N}} \right|^2 = |\hat{f}(0)|^2$$

which proves the lemma. \square

It is worth noting that if the eigenvector matrix Φ has all constant-amplitude entries, i.e., $|\Phi_{ij}| = 1/\sqrt{N}$, then graph translation is isometric. An example of such a graph is the cycle graph on N vertices, C_N , if we allow Φ to equal the complex-valued DFT matrix. Then since $|\varphi_k(n)| = 1/\sqrt{N}$ for all $k = 0, \dots, N-1$

and $n = 1, \dots, N$ the calculation in the proof of Lemma 2.8 gives

$$\|T_i f\|_2^2 = N \sum_{k=0}^{N-1} |\hat{f}(\lambda_k)|^2 |\varphi_k^*(i)|^2 = \sum_{k=0}^{N-1} |\hat{f}(\lambda_k)|^2 = \|f\|_2^2,$$

where the last equality follows from Parseval's identity (Proposition 2.1.2).

Furthermore, unlike the Euclidean notion of translation, graph translation need not be invertible. Theorem 2.4.8 characterizes all graphs for which the operator T_i is not invertible.

Theorem 2.4.8. *The graph translation operator T_i fails to be invertible if and only if there exists some $k = 1, \dots, N-1$ for which $\varphi_k(i) = 0$. Furthermore, the nullspace of T_i has a basis equal to those eigenvectors that vanish on the i 'th vertex.*

Proof. Assuming some $\varphi_k(i) = 0$, we prove that T_i is not invertible by calculating its rank. By (2.7), the operator T_i can be written as the matrix

$$\begin{aligned} T_i &= \sqrt{N} \begin{pmatrix} \varphi_0^*(i)\varphi_0(1) & \varphi_1^*(i)\varphi_1(1) & \cdots & \varphi_{N-1}^*(i)\varphi_{N-1}(1) \\ \vdots & \vdots & \cdots & \vdots \\ \varphi_0^*(i)\varphi_0(N) & \varphi_1^*(i)\varphi_1(N) & \cdots & \varphi_{N-1}^*(i)\varphi_{N-1}(N) \end{pmatrix} \Phi^* \\ &=: \sqrt{N} A_i \Phi^* \end{aligned} \tag{2.9}$$

We can compute $T_i^* T_i = N \Phi A_i^* A_i \Phi^*$. Since Φ is an $N \times N$ matrix of full rank, we can express the rank of T_i solely in terms of the matrix A_i , i.e.,

$$\text{rank}(T_i) = \text{rank}(T_i^* T_i) = \text{rank}(\Phi A_i^* A_i \Phi^*) = \text{rank}(A_i^* A_i).$$

We can explicitly compute for any indices $n, m \in \{1, \dots, N\}$,

$$\begin{aligned}
(A_i^* A_i)(n, m) &= \sum_{k=1}^N \overline{A_i(k, n)} A_i(k, m) = \sum_{k=0}^{N-1} \varphi_n^*(k) \varphi_n(i) \varphi_m(k) \varphi_m^*(i) \\
&= \varphi_n(i) \varphi_m^*(i) \sum_{k=0}^{N-1} \varphi_n^*(k) \varphi_m(k) \\
&= \varphi_n(i) \varphi_m^*(i) \delta_n(m).
\end{aligned}$$

Hence, $A_i^* A_i$ is a diagonal matrix with diagonal entries $(A_i^* A_i)(n, n) = |\varphi_n(i)|^2$ and therefore

$$\text{rank}(T_i) = |\{k : \varphi_k(i) \neq 0\}|.$$

Conversely, if T_i is not invertible then it must not be of full rank and therefore at least one entry on the diagonal matrix $A_i^* A_i$ must equal zero. This proves the first part of the theorem.

To prove the second part of the theorem, suppose that $\varphi_{k_j}(i) = 0$ for $\{k_j\}_{j=1}^K \subseteq \{1, \dots, N-1\}$ (hence, $K \leq N-1$). Then for each $j \in \{1, \dots, K\}$ and any $n \in \{1, \dots, N\}$ we have

$$T_i \varphi_{k_j}(n) = \sqrt{N} \sum_{k=0}^{N-1} \hat{\varphi}_{k_j}(\lambda_k) \varphi_k^*(i) \varphi_k(n) = \sqrt{N} \varphi_{k_j}^*(i) \varphi_{k_j}(n) = 0.$$

Therefore, φ_{k_j} is in the null space of T_i for every $j = 1, \dots, K$. Moreover they are $\{\varphi_{k_j}\}_{j=1}^K$ is a collection of K orthogonal unit-norm vectors in the null space which has dimension $N - \text{rank}(T_i) = K$, hence they form an orthonormal basis for the null space of T_i . \square

Corollary 2.4.9. *If $\varphi_k(i) \neq 0$ for all $k = 1, \dots, N-1$, then the graph translation*

operator T_i is invertible and its inverse is given by

$$T_i^{-1} = \frac{1}{\sqrt{N}} \Phi \begin{pmatrix} \varphi_0^*(1)\varphi_0^*(i)^{-1} & \varphi_0^*(2)\varphi_0^*(i)^{-1} & \cdots & \varphi_0^*(N)\varphi_0^*(i)^{-1} \\ \vdots & \vdots & \cdots & \vdots \\ \varphi_{N-1}^*(1)\varphi_{N-1}^*(i)^{-1} & \varphi_{N-1}^*(2)\varphi_{N-1}^*(i)^{-1} & \cdots & \varphi_{N-1}^*(N)\varphi_{N-1}^*(i)^{-1} \end{pmatrix}.$$

Proof. We shall first prove that the inverse to the matrix A_i given in (2.9) is given

by

$$A_i^{-1} = \begin{pmatrix} \varphi_0^*(1)\varphi_0^*(i)^{-1} & \varphi_0^*(2)\varphi_0^*(i)^{-1} & \cdots & \varphi_0^*(N)\varphi_0^*(i)^{-1} \\ \vdots & \vdots & \cdots & \vdots \\ \varphi_{N-1}^*(1)\varphi_{N-1}^*(i)^{-1} & \varphi_{N-1}^*(2)\varphi_{N-1}^*(i)^{-1} & \cdots & \varphi_{N-1}^*(N)\varphi_{N-1}^*(i)^{-1} \end{pmatrix}.$$

We can then compute

$$\begin{aligned} A_i A_i^{-1}(n, m) &= \sum_{k=1}^N A_i(n, k) A_i^{-1}(k, m) = \sum_{k=0}^{N-1} \varphi_k^*(i) \varphi_k(n) \varphi_k^*(m) \varphi_k^*(i)^{-1} \\ &= \sum_{k=0}^{N-1} \varphi_k(n) \varphi_k^*(m) = \delta_n(m), \end{aligned}$$

and similarly

$$\begin{aligned} A_i^{-1} A_i(n, m) &= \sum_{k=1}^N A_i^{-1}(n, k) A_i(k, m) = \sum_{k=1}^N \varphi_{n-1}^*(k) \varphi_{n-1}^*(i)^{-1} \varphi_{m-1}(k) \varphi_{m-1}^*(i) \\ &= \varphi_{n-1}^*(i)^{-1} \varphi_{m-1}^*(i) \sum_{k=1}^N \varphi_{n-1}^*(k) \varphi_{m-1}(k) = \varphi_{n-1}^*(i)^{-1} \varphi_{m-1}^*(i) \delta_n(m), \end{aligned}$$

which proves $A_i^{-1} A_i = A_i A_i^{-1} = I_N$.

Thus we can verify by the orthonormality of Φ that

$$T_i T_i^{-1} = A_i \Phi^* \Phi A_i^{-1} = I_N = \Phi A_i^{-1} A_i \Phi^* = T_i^{-1} T_i.$$

□

Since the invertability of the graph translation operators depends entirely on when and where eigenvectors vanish, Chapter 3 is devoted to studying the support of graph eigenvectors.

Remark 2.4.10. The results of Theorem 2.4.8 and its corollary are not applicable solely to the graph translation operators. They can be generalized to a broader class of operators on graphs, in particular, operators that act as Fourier operators. We say that an operator A that acts on a function f is a Fourier multiplier if the Fourier transform of Af can be written as the product

$$\widehat{Af}(\xi) = \hat{a}(\xi)\hat{f}(\xi)$$

for some function a defined in the spectral domain.

Indeed graph translation is defined as a Fourier multiplier since it is defined as

$$\widehat{T_i f}(\lambda_k) = \varphi_k(i)\hat{f}(\lambda_k).$$

Hence, Theorem 2.4.8 and Corollary 2.4.9 can be generalized to Fourier multipliers in the following way

Corollary 2.4.11. *Let A be a Fourier multiplier whose action on $f : V \rightarrow \mathbb{C}$ is defined in the spectral domain $\widehat{Af}(\lambda_k) = \hat{a}(\lambda_k)\hat{f}(\lambda_k)$. Then A is invertible if and only if $\hat{a}(\lambda_k) \neq 0$ for all $\lambda = 0, 1, \dots, N - 1$. Furthermore, its inverse A^{-1} will be given by the Fourier multiplier*

$$\widehat{A^{-1}f}(\lambda_k) = \hat{a}(\lambda_k)^{-1}\hat{f}(\lambda_k).$$

Chapter 3

Support of eigenvectors on graphs

This chapter proves results about the support of Laplacian eigenvectors on graphs. In particular, we characterize and describe the set on which eigenvectors vanish. The Fiedler vector, φ_1 , has unique properties that enable us to prove our main result, Theorem 3.1.10, that planar graphs cannot have large regions on which φ_1 vanishes. We then construct a family of (non-planar) graphs, called the barren graphs, and prove in Theorem 3.1.13 that their Fiedler vectors do vanish on large regions.

3.1 Support and the Characteristic set of the Fiedler Vector

A connected $G(V, E, \omega)$ graph on N vertices has Laplacian L with real spectrum $0 = \lambda_0 < \lambda_1 \leq \dots \leq \lambda_{N-1}$ and corresponding real orthonormal eigenvectors $\varphi_0, \varphi_1, \dots, \varphi_{N-1}$. By Theorem 1.2.1, the first eigenfunction, φ_0 , will be the constant vector $1/\sqrt{N}$. The first nontrivial eigenvalue, λ_1 , is of particular importance. It is a measure of how connected the graph G is, and as such it is known as the algebraic connectivity of the graph G . The highest λ_1 can be is N , which happens only for the complete graph in which case the spectrum is $\{0, N, \dots, N\}$.

An eigenvector corresponding to the algebraic connectivity of G is called the *Fiedler vector*, φ_1 , satisfying $L\varphi_1 = \lambda_1\varphi_1$, named after Czech mathematician

Miroslav Fiedler and his contributions to this subject [29,30]. If λ_1 has multiplicity 1, then the Fiedler vector is unique up to a sign. The Fiedler vector is used extensively in dimensionality reduction techniques [5, 19, 20, 22], data clustering [54], image segmentation [28], and graph drawing [60]. As seen in Theorem 2.4.8, the support of eigenvectors will influence the behavior of the graph translation operators defined in the emerging field of graph signal processing literature. The goal of this chapter is to characterize where a Fiedler vector, and other eigenvectors of the Laplacian, vanish on V .

Let φ_1 denote a Fiedler vector for L on G . We can decompose the vertex set, V , into three disjoint subsets, $V = V_+ \cup V_- \cup V_0$, where $V_+ = \{x \in V : \varphi_1(x) > 0\}$, $V_- = \{x \in V : \varphi_1(x) < 0\}$, and $V_0 = \{x \in V : \varphi_1(x) = 0\}$. The set V_0 , the set of vertices on which the Fiedler vector vanishes, is referred to in literature as the *characteristic set* of the graph [2]. The characteristic set is sometimes referred to as the nodal vertices of the graph in some literature. This vertex decomposition is not a unique property to the graph G ; any graph can allow multiple such decompositions of the vertex set V . In the case that the algebraic connectivity has higher multiplicities, i.e., $\lambda_1 = \lambda_2 = \dots = \lambda_m$ for some $2 \leq m \leq N - 1$, then each φ_s is a Fiedler vector for $1 \leq s \leq m$. Furthermore, any linear combination of $\{\varphi_s\}_{s=1}^m$ will also be a Fiedler vector and yield a different vertex decomposition. Even in the case when the algebraic connectivity of G is simple, then $-\varphi_1$ is also a Fiedler vector for G . In this case, V_+ and V_- can be interchanged but the set V_0 is unique to G . It was recently shown in [68] that the space of Hermitian matrices (which includes graph Laplacians) with simple spectrum is open and dense in the space of all Hermitian

matrices.

We wish to describe and characterize the sets V_+ , V_- , and V_0 for graphs. Fiedler proved in [30] that the subgraph induced on the vertices $\{v \in V : \varphi_1(v) \geq 0\} = V_+ \cup V_0$ forms a connected subgraph of G . Similarly, $V_- \cup V_0$ form a connected subgraph of G . Recently, it was proved in [69] that we can relax the statement and show that the subgraphs on V_+ and V_- are connected subgraphs of G .

The following result guarantees that V_+ and V_- are always close in terms of the shortest path graph distance.

Lemma 3.1.1. *Let $G(V, E, \omega)$ with Fiedler vector φ_1 inducing the partition of vertices $V = V_+ \cup V_- \cup V_0$. Then $d(V_+, V_-) \leq 2$.*

Proof. First consider the case in which $V_0 = \emptyset$. In this case, there necessarily exists an edge $e = (x, y)$ with $x \in V_+$ and $y \in V_-$ and hence $d(V_+, V_-) = 1$.

Now consider the case in which $V_0 \neq \emptyset$. Since G is connected we are guaranteed the existence of some $x \in V_0$ and some $y \sim x$ with either $y \in V_+$ or $y \in V_-$. Since $x \in V_0$, we have

$$0 = \lambda_1 \varphi_1(x) = L\varphi_1(x) = \sum_{z \sim x} \varphi_1(z) - \varphi_1(x) = \sum_{z \sim x} \varphi_1(z). \quad (3.1)$$

We have established that at least one neighbor of x , namely y , satisfied $\varphi_1(y) \neq 0$. Then by (3.1), there must exist at least one other neighbor of x , call it y' , such that $\varphi_1(y')$ has the opposite sign of $\varphi_1(y)$. Hence we have now constructed a path, namely (y, x, y') connecting V_+ and V_- and the lemma is proved. \square

Many graphs exhibit the property that eigenvectors φ_k for large values of k are highly localized and vanish on large regions of the graph; see [37] for an experimental

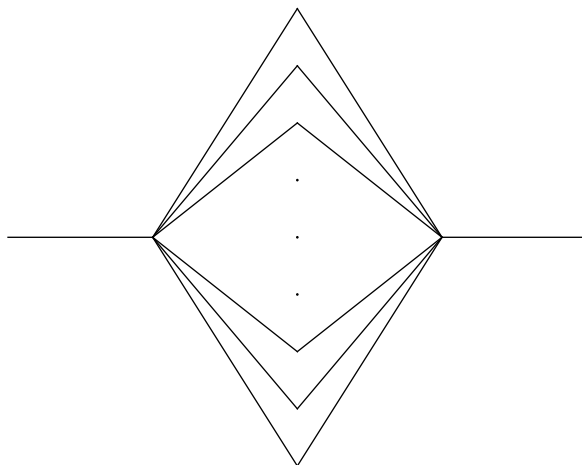


Figure 3.1: A graph with arbitrarily large set $V_0(\varphi)$

excursion on this phenomenon. It is perhaps a misconception that eigenvectors corresponding to small eigenvalues, or in particular, the Fiedler vector of graphs have full support. Indeed the Fiedler vector of the Minnesota graph never achieves value zero. On the other hand the Fiedler vector of the the graph approximations to the Sierpeinski gastet SG_n can vanish but only along the small number of vertices symmetrically in the center of the graph.

It was shown in [2], that the cardinality of V_0 can be arbitrarily large. Figure 3.1 shows a family of graphs that yield sets V_0 with arbitrarily large cardinality. The family is a path graph P_N on an odd number of vertices, except the middle vertex and its edges are duplicated an arbitrarily large number of times. As evident from Figure 3.1, the set V_0 is not connected; in fact, no vertex in V_0 is connected to any other vertex of V_0 .

For the sake of thoroughness we introduce a family of graphs also with arbitrar-

ily large V_0 but that is also connected. We call the family of graphs the *generalized ladder graphs*, denoted $\text{Ladder}(n, m)$. The standard ladder graphs, $\text{Ladder}(n, 2)$, is simply the graph Cartesian product, see [39], of the path graph of length n , P_n , and the path graph of length 1, P_1 . The graph $\text{Ladder}(n, 2)$ resembles a ladder with n rungs. The generalized ladder graphs, $\text{Ladder}(n, m)$, are ladders with n rungs and each rung contains m vertices. Provided that the number of rungs, n , is odd, then V_0 will be the middle rung and will clearly be connected. This gives $|V_0| = m$. Figure 3.2 shows a generalized ladder graph and its Fiedler vector.

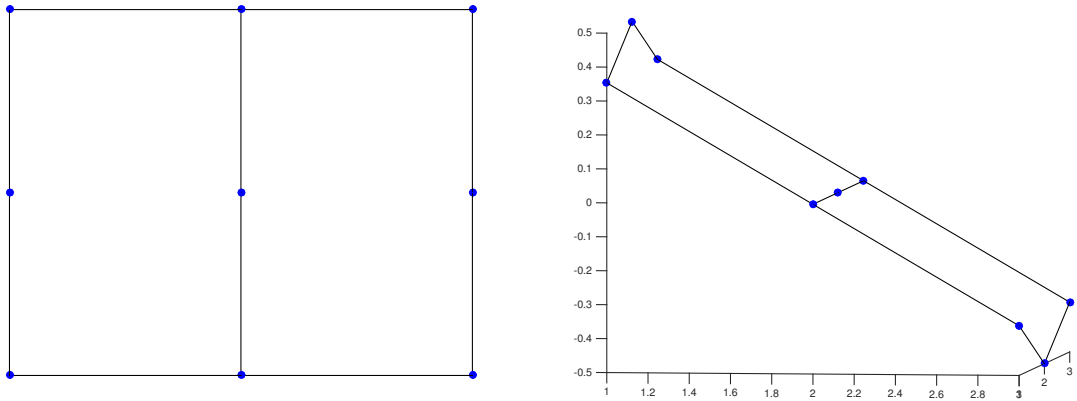


Figure 3.2: The generalized ladder graph, $\text{Ladder}(3, 3)$. The set V_0 consists of the three vertices making the middle rung of the ladder and contains one ball of three vertices.

The generalized ladder graph provides an example of a graph with a connected characteristic set. Observe in Figure 3.2 however, that each vertex in V_0 is connected to at most two vertices. We then pose the question as to whether or not there exist graphs for which a vertex V_0 has three or more neighbors all contained in V_0 . It is

simpler to state this property using the definition of a graph ball, which is motivated by the definition of a closed ball in general normed spaces.

Definition 3.1.2. Given any $x \in V$ and any integer $r \geq 1$, we define the *ball of radius r centered at x* ,

$$B_r(x) = \{y \in V : d(x, y) \leq r\},$$

where $d(x, y)$ is the shorted path length from x to y in G .

The following proposition shows that the Fiedler vector cannot be constant-valued on any balls within V_+ and V_- .

Proposition 3.1.3. *Let φ_1 be the Fiedler vector for the Laplacian on graph G and suppose $B_r(x) \subseteq V_+$ or $B_r(x) \subseteq V_-$ for $r \geq 1$. Then φ_1 cannot be constant-valued on $B_r(x)$.*

Proof. It suffices to prove the claim for $r = 1$. Without loss of generality, assume $B_1(x) \subseteq V_+$ and suppose that φ_1 is constant on $B_1(x)$. Then

$$L\varphi_1(x) = \sum_{y \sim x} \omega_{x,y}(\varphi_1(x) - \varphi_1(y)) = 0,$$

since $y \sim x$ implies $y \in B_1(x)$ and φ_1 is constant on that ball. However, $L\varphi_1(x) = \lambda_1\varphi_1(x) > 0$ since $\lambda_1 > 0$ and $\varphi_1(x) > 0$ on V_+ . This is a contradiction and the proof is complete. \square

The result of Proposition 3.1.3 can be formulated in terms on *any* nonconstant eigenvector of the Laplacian, not just a Fiedler vector.

Corollary 3.1.4. *Any nonconstant eigenvector of the Laplacian, φ_k , associated with eigenvalue $\lambda_k > 0$ cannot be constant on any ball contained in the positive vertices $\{i \in V : \varphi_k(i) > 0\}$ or negative vertices $\{i \in V : \varphi_k(i) < 0\}$ associated to that eigenvector.*

Proof. Suppose there existed a ball $B_1(x) \subseteq \{i \in V : \varphi_k(i) > 0\}$ on which φ_k was constant. Then just as in the previous proof we could calculate

$$L\varphi_k(x) = \sum_{y \sim x} \omega_{x,y}(\varphi_k(x) - \varphi_k(y)) = 0,$$

which contradicts $L\varphi_k(x) = \lambda_k\varphi_k(x) > 0$. □

We wish to extend Proposition 3.1.3 to the set V_0 . However, as seen in generalized ladder graphs, $\text{Ladder}(n, m)$ for n odd and $m > 2$, for which V_0 contains a ball of radius 1. This ball, however, contains 3 vertices (the center vertex and its two neighbors). The next goal is to characterize graphs whose characteristic set V_0 contains a ball of radius 1 containing at least four vertices. We prove that this is impossible for planar graphs.

Definition 3.1.5. A *planar graph* is a graph whose vertices and edges can be embedded in \mathbb{R}^2 with edges intersecting only at vertices.

In 1930, Kazimierz Kuratowski characterized all planar graphs in terms of subdivisions.

Definition 3.1.6. A *subdivision* of a graph $G(V, E)$, also referred to as an *expansion*, is the graph $H(\tilde{V}, \tilde{E})$ where the vertex set is the original vertex set with an added

vertex, w , and the edge set replaces an edge (u, v) with the two edges (u, w) and (w, v) . That is, $\tilde{V} = V \cup \{w\}$ and $\tilde{E} = E \setminus \{(u, v)\} \cup \{(u, w), (w, v)\}$.

Theorem 3.1.7 (Kuratowski's Theorem, [42]). *A finite graph, G , is planar if and only if it does not contain a subgraph that is a subdivision of K_5 or $K_{3,3}$, where K_5 is the complete graph on 5 vertices and $K_{3,3}$ is the complete bipartite graph on six vertices (also known as the utility graph), see Figure 3.3.*

A weaker formulation of Kuratowski's Theorem can be stated in terms of graph minors.

Definition 3.1.8. Given an undirected graph $G(V, E)$, consider edge $e = (u, v) \in E$. Contracting the edge e entails deleting edge e and identifying u and v as the same vertex. The resulting graph $H(\tilde{V}, \tilde{E})$ has one fewer edge and vertex as G .

An undirected graph is called a *minor* of G if it can be formed by contracting edges of G .

Theorem 3.1.9 (Wagner's Theorem, [71]). *A finite graph is planar if and only if it does not have K_5 or $K_{3,3}$ as a minor.*

Because of the importance of K_5 and $K_{3,3}$ in identifying non-planar graphs, there are referred to as *forbidden minors*.

One of the main results in this chapter shows that planar graphs cannot have large balls contained in the characteristic set V_0 .

Theorem 3.1.10. *Let $G(V, E)$ be a planar graph with Fiedler vector φ_1 . Then the zero set of φ_1 contains no balls of radius $r = 1$ with more than three vertices.*

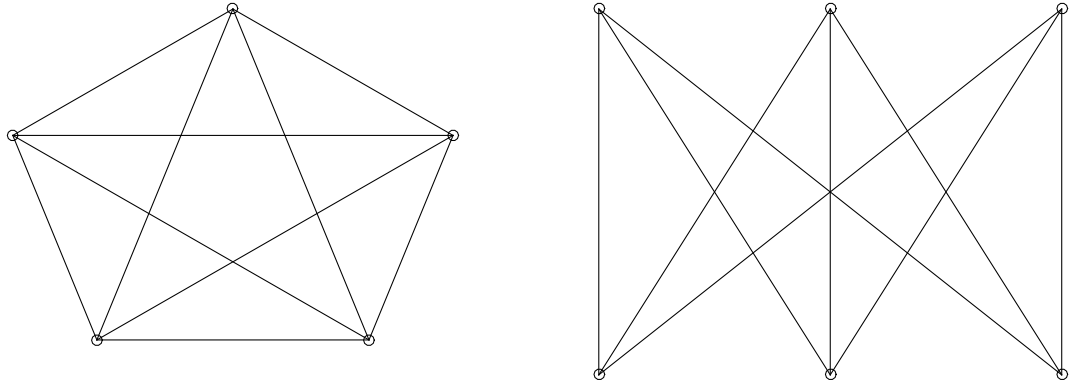


Figure 3.3: The forbidden minors. Left: The complete graph on five vertices. Right: The complete bipartite graph on six vertices.

Proof. Suppose that V_0 contains a ball, $B_1(x)$, centered at vertex $x \in V_0$ and comprised of at least four vertices. Without loss of generality, we can assume that the connected component of V_0 containing x equals $B_1(x)$. If not, then contract edges so that the connected component of V_0 containing x equals a ball of radius 1. Since $|B_1(x)| \geq 4$, then we have $d_x \geq 3$ and let $\{y_i\}_{i=1}^{d_x}$ denote the neighbors of x . Then as constructed, $B_1(x) = \{x, y_1, y_2, \dots, y_{d_x}\}$.

By Lemma 3.1.1, for $i = 1, 2, 3$, each vertex y_i has at least one neighbor in V_+ and at least one in V_- ; pick one neighbor from each set and denote them p_i and n_i , respectively. It is proved in [69] that V_+ and V_- are connected subgraphs of G . Therefore, there is a path of edges that connect p_1, p_2 , and p_3 (if $p_1 = p_2 = p_3$, then this path is empty). We create a minor of G by contracting the path connecting p_1, p_2 , and p_3 to create one vertex $p \in V_+$. Similarly, since V_- is connected, we can contract the path connecting n_1, n_2 , and n_3 , to create one vertex $n \in V_-$.

Consider the subgraph of the now minorized version of G consisting of vertices $\{x, p, n, y_1, y_2, y_3\}$. This subgraph is $K_{3,3}$, the complete bipartite graph on six vertices since the vertices $\{x, p, n\}$ are all connected to $\{y_1, y_2, y_3\}$. Thus by Wagner's Theorem, G is not a planar graph, which is a contradiction. This completes the proof. \square

The result of Theorem 3.1.10 does not hold for general graphs. We construct a family of (nonplanar) graphs for which V_0 contains a ball with a large number of vertices. Since the set of vertices for which the Fiedler vector vanishes is large, we call this family of graphs the *barren graphs*. The barren graph with $|V| = N + 7$ and $|V_0| = N + 1$ is denoted $\text{Barr}(N)$.

3.1.1 Construction of the barren graph, $\text{Barr}(N)$

The barren graph will be constructed as a sum of smaller graphs.

Definition 3.1.11. Let $G_1(V, E_1)$ and $G_2(V, E_2)$ be two graphs. The *sum* of graphs G_1 and G_2 is the graph $G(V, E)$ where $E = E_1 \cup E_2$.

The barren graph $\text{Barr}(N)$ is defined as follows

Definition 3.1.12. Let $K(V_i, V_j)$ denote the bipartite complete graph between vertex sets V_i and V_j , that is, the graph with vertex set $V = V_i \cup V_j$ and edge set $E = \{(x, y) : x \in V_i, y \in V_j\}$. For $N \geq 3$ the barren graph, $\text{Barr}(N)$, is a graph with $N + 7$ vertices. Let $\{V_i\}_{i=1}^6$ denote distinct vertex sets with given cardinalities $\{|V_i|\}_{i=1}^6 = \{N, 1, 2, 2, 1, 1\}$. Then the barren graph is the following graph sum of

the 5 complete bipartite graphs

$$\text{Barr}(N) = K(V_1, V_2) + K(V_1, V_3) + K(V_1, V_4) + K(V_3, V_5) + K(V_4, V_6).$$

As constructed, $\text{Barr}(N)$ itself is bipartite; all edges connect the sets $V_2 \cup V_3 \cup V_4$ to $V_1 \cup V_5 \cup V_6$. Figure 3.4 shows two examples of barren graphs.

We shall show that for any N , the Fiedler vector for $\text{Barr}(N)$ vanishes on $V_1 \cup V_2$ which has cardinality $N + 1$. Hence, the Fiedler vector for $\text{Barr}(N)$ has support on exactly six vertices for any $N \geq 3$. In order to prove this, we explicitly derive the entire spectrum and all eigenvectors of the Laplacian.

Theorem 3.1.13. *The barren graph, $\text{Barr}(N)$, has the spectrum given in Table 3.1. In particular, the Fiedler vector of $\text{Barr}(N)$ vanishes on vertices $V_1 \cup V_2$ and hence $|\text{supp}(\varphi_1)| = 6$ for any N .*

Proof. Firstly, the graph $\text{Barr}(N)$ is connected and so we have $\lambda_0 = 0$ with $\varphi_0 \equiv (N + 7)^{-1/2}$. All other eigenvalues must be positive.

We will next show that the structure and support of the function shown in Figure 3.5 is an eigenvector for two eigenvalues of $\text{Barr}(N)$. One can check upon inspection that the shown function φ is orthogonal to the constant function. Then, if the function shown in Figure 3.5, call it φ , is an eigenvector, then the eigenvalue equation, $Lx = \lambda x$ is satisfied at each vertex. It is easy to verify that $L\varphi(x) = 0$ for each $x \in V_1 \cup V_2$. For $x \in V_5$ or $x \in V_6$ the eigenvalue equation becomes $L\varphi(x) = 2(b - a) = \lambda b$. Finally, for any $x \in V_3$ or $x \in V_4$, the eigenvalue equation gives $L\varphi(x) = Na + (a - b) = \lambda a$. Finally, we also impose that the condition that

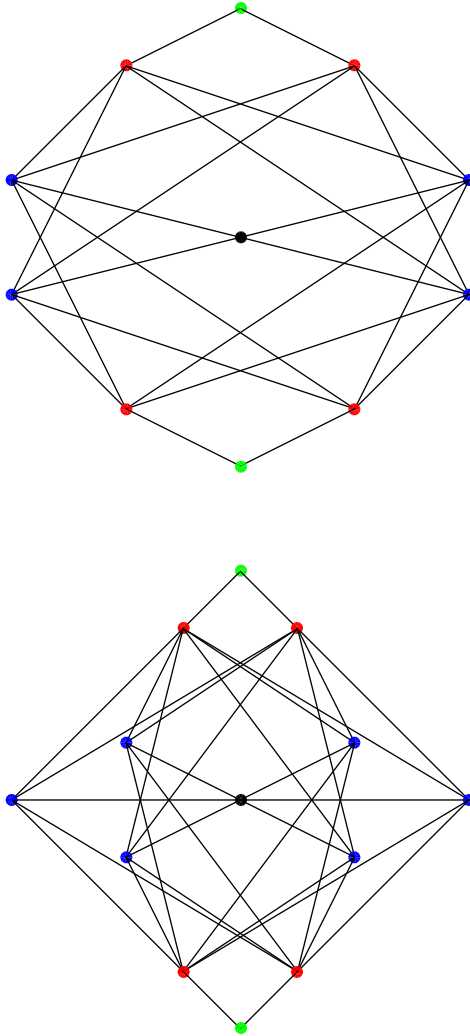


Figure 3.4: The barren graph Barr(4) (top) and Barr(6) (bottom).

The set V_1 is denoted with N blue dots; the vertex set V_2 is the black vertex in the center; the sets V_3 and V_4 are denoted with red dots; the sets V_5 and V_6 are denoted with green dots.

λ_k	value	eigenvector
λ_0	0	constant function
λ_1	$\frac{1}{2}(N+3-\sqrt{N^2-2N+9})$	Figure 3.5
λ_2	y_1	Figure 3.6
$\lambda_3 = \dots = \lambda_{N+1}$	5	ON basis on V_1
λ_{N+2}	y_2	Figure 3.6
$\lambda_{N+3} = \lambda_{N+4}$	$N+1$	Figure 3.7
λ_{N+5}	$\frac{1}{2}(N+3+\sqrt{N^2-2N+9})$	Figure 3.5
λ_{N+6}	y_3	Figure 3.6

Table 3.1: The spectrum of the barren graph, $\text{Barr}(N)$. The values y_1, y_2, y_3 are the roots to the cubic polynomial (3.2).

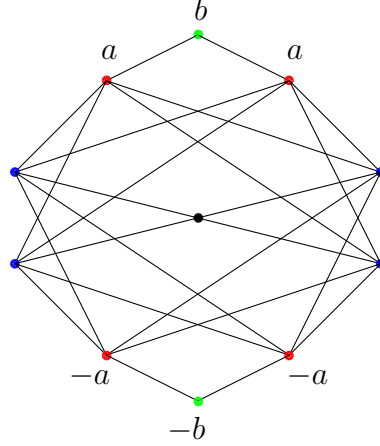


Figure 3.5: Support and function values for the eigenvectors associated with eigenvalues λ_1 and λ_{N+5} .

the eigenvectors are normalized so that $\|\varphi\| = 1$. Therefore, the function φ shown in Figure 3.5 is an eigenvector of L if and only if the following system of equations has a nontrivial solution:

$$\begin{cases} 4a^2 + 2b^2 = 1 \\ 2(b - a) = \lambda b \\ Na + (a - b) = \lambda a. \end{cases}$$

The first equation is not linear, but we can still solve this system by hand with substitution to obtain the following two solutions:

$$\begin{cases} a = \frac{1}{2} \sqrt{\frac{N^2 - 2N + 9 \mp (N-1)\sqrt{N^2 - 2N + 9}}{2(N^2 - 2N + 9)}} \\ b = \frac{1}{2} \sqrt{\frac{N^2 - 2N + 9 \pm (N-1)\sqrt{N^2 - 2N + 9}}{N^2 - 2N + 9}} \\ \lambda = \frac{1}{2} (N + 3 \pm \sqrt{N^2 - 2N + 9}). \end{cases}$$

This gives two orthogonal eigenvectors and their eigenvalues.

Consider now the vector shown in Figure 3.6 with full support, yet only taking on four distinct values. Similar to the previous example, we obtain a system of

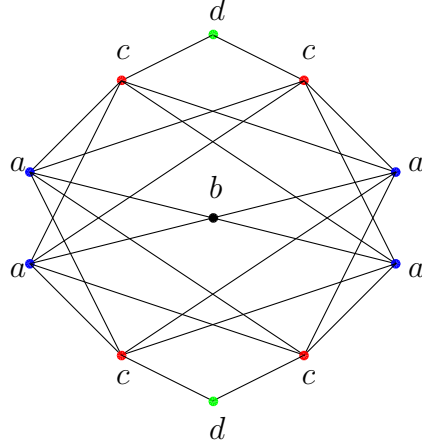


Figure 3.6: Support and function values for the eigenvectors associated with eigenvalues λ_2 , λ_{N+2} , and λ_{N+6} .

equations by imposing the conditions $\|\varphi\| = 1$, $\langle \varphi, 1 \rangle = 0$, and from writing out the eigenvalue equations at each vertex class from V_1, V_2, V_3 and V_5 which gives:

$$\left\{ \begin{array}{ll} Na^2 + b^2 + 4c^2 + 2d^2 = 1 & (\|\varphi\|^2 = 1) \\ Na + b + 4c + 2d = 0 & (\varphi \perp 1) \\ 4(a - c) + (a - b) = \lambda a & (L\varphi(x) = \lambda\varphi(x) : x \in V_1) \\ N(b - 1) = \lambda b & (L\varphi(x) = \lambda\varphi(x) : x \in V_2) \\ (c - d) + N(c - a) = \lambda c & (L\varphi(x) = \lambda\varphi(x) : x \in V_3 \cup V_4) \\ 2(d - c) = \lambda d & (L\varphi(x) = \lambda\varphi(x) : x \in V_5 \cup V_6) \end{array} \right. .$$

Again, this system cannot be solved with linear methods. However, by tedious substitutions we can reduce the system (assuming each of the variables a, b, c, d, λ are nonzero) to solving for the roots of the following cubic polynomial in λ :

$$\lambda^3 + (-2N - 8)\lambda^2 + (N^2 + 10N + 15)\lambda + (-2N^2 - 14N) = 0 \quad (3.2)$$

The cubic polyomial $x^3 + c_2x^2 + c_1x + c_0 = 0$ has three distinct real roots if its

discriminant, $\Delta = 18c_0c_1c_2 - 4c_2^3c_0 + c_2^2c_1^2 - 4c_1^3 - 27c_0^2$, is positive. The discriminant of (3.2) is positive for all $N > 0$ and hence we let $y_1 < y_2 < y_3$ denote the three positive roots which make up λ_2, λ_{N+2} , and λ_{N+6} , respectively. By substituting back into the system of equations, one can obtain values for a, b, c, d for each of the $\lambda = y_1, y_2, y_3$.

The roots y_1, y_2, y_3 monotonically increase in N . A simple calculation shows that $y_1 = 2$ for $N = 3$ and $y_1 > 2$ for $N > 3$. Hence $\lambda_1 < \lambda_2 = y_1$ for all N . Also observe that $y_2 < 5$ for $N < 5$, so the ordering of the eigenvalues in Table 3.1 can vary but their values are accurate.

One can verify that the three eigenvectors obtained from Figure 3.6 are linearly independent and orthogonal to each eigenvector derived so far.

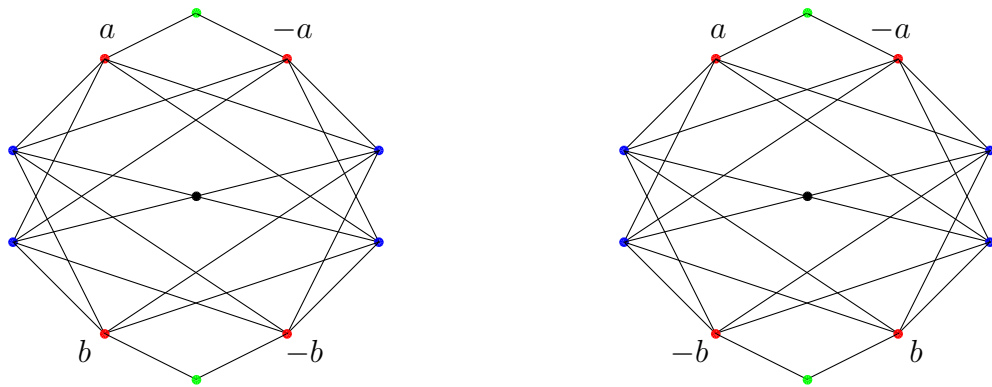


Figure 3.7: Support and function values for the eigenvectors associated with eigenvalues λ_{N+3} and λ_{N+4} .

Consider now the two functions shown in Figure 3.7. The eigenvalue equation gives $L\varphi(x) = 0$ for every $x \in V_3 \cup V_4$ in which case we have

$L\varphi(x) = (N + 1)\varphi(x)$. The two functions shown in Figure 3.7 are orthogonal and linearly independent to each other and every eigenvector derived thus far and hence $N + 1$ is an eigenvalue of $\text{Barr}(N)$ with multiplicity two.

Finally, we will construct eigenfunctions that are supported only on the N vertices in V_1 . Observe that if a function, f , is supported on V_1 then for any $x \in V_1$, the eigenvalue equation gives $Lf(x) = 5f(x)$ since x neighbors five vertices on which f vanishes. Therefore, $\text{Barr}(N)$ has eigenvalue 5. To construct the corresponding eigenbasis, one can choose any orthonormal basis for the subspace of the N -dimensional vector space that is orthogonal to the constant vector. Any basis for this $(N - 1)$ -dimensional vector space will give an eigenbasis for on V_1 . Finally one can verify by inspection that these $N - 1$ eigenvectors are orthogonal and linearly independent to each eigenvector derived in this proof.

As such, we have now constructed an orthonormal, linearly independent eigenbasis for $\text{Barr}(N)$ corresponding to the eigenvalues given in Table 3.1. \square

As a remark, observe the behavior of the spectrum of $\text{Barr}(N)$ as $N \rightarrow \infty$. For every natural number N , $\lambda_1 < 2$ and $\lim_{N \rightarrow \infty} \lambda_1 = 2$. Using a symbolic solver, one can prove that $\lim_{N \rightarrow \infty} \lambda_2 = \lim_{N \rightarrow \infty} y_1 = 2$ as well. Furthermore, the other two roots of the polynomial (3.2) tend to infinity as $N \rightarrow \infty$. Therefore, as $N \rightarrow \infty$, $\text{Barr}(N)$ has spectrum approaching 0 (with multiplicity 1), 2 (with multiplicity 2), 5 (with multiplicity $N - 2$), and the rest of the eigenvalues tending to ∞ .

3.1.2 Characteristic Vertices and graph adding

In this subsection, we prove results about eigenvectors and their characteristic vertices for graph sums as defined in Definition 3.1.11. We borrow the following notation from [69] for clarity.

Definition 3.1.14. For any function f , we make the following definitions,

$$i_0(f) = \{i \in V : f(i) = 0\},$$

$$i_+(f) = \{i \in V : f(i) > 0\},$$

$$i_-(f) = \{i \in V : f(i) < 0\}.$$

Observe that the set V_0 (resp. V_+ and V_-) from Section 3.1 is equal to $i_0(\varphi_1)$ (resp. $i_+(\varphi_1)$ and $i_-(\varphi_1)$).

Theorem 3.1.15. Consider $n \geq 2$ connected graphs, $\{G_j(V_j, E_j)\}_{j=1}^n$ and suppose that all n graph Laplacians, L_j , share a common eigenvalue $\lambda > 0$ with corresponding eigenvectors $\varphi_{(j)}$. Each graph's vertex set, V_j , assumes a decomposition $V_j = i_+(\varphi_{(j)}) \cup i_-(\varphi_{(j)}) \cup i_0(\varphi_{(j)})$ and suppose that $i_0(\varphi_{(j)}) \neq \emptyset$ for all j . Consider the graph $G(V, E) = G(\cup_{j=1}^n V_j, \cup_{j=1}^n E_j \cup E_0)$ where the edge set $E_0 = \{(x_i, y_i)\}_{i=1}^K$ for $x_i \in i_0(\varphi_j)$, $y_i \in i_0(\varphi_\ell)$, and $j \neq \ell$ is nonempty. Define φ on G by $\varphi(x) = \varphi_{(j)}(x)$ for $x \in V_j$. Then, λ is an eigenvalue of G and φ is a corresponding eigenvector.

Proof. We will verify that $L\varphi(x) = \lambda\varphi(x)$ for every $x \in V$. Every $x \in V$ lies in exactly one V_j and every edge connecting to x must be in either E_j or E_0 . Suppose x contains no edges from E_0 . Then $L\varphi(x) = L_j\varphi_{(j)}(x) = \lambda\varphi_{(j)}(x) = \lambda\varphi(x)$.

Suppose instead that x does contain at least one edge from E_0 . Then by construction of the set E_0 , we have $\varphi(x) = 0$ and $\varphi(y) = 0$ for all $(x, y) \in E_0$. This allows us to compute

$$\begin{aligned} L\varphi(x) &= \sum_{y \sim x} (\varphi(x) - \varphi(y)) = \sum_{\substack{y \sim x \\ (x,y) \in E_j}} (\varphi(x) - \varphi(y)) + \sum_{\substack{y \sim x \\ (x,y) \in E_0}} (\varphi(x) - \varphi(y)) \\ &= L_j \varphi_j(x) + 0 = \lambda \varphi_j(x) = \lambda \varphi(x). \end{aligned}$$

Hence for any vertex in V , the vector φ satisfies the eigenvalue equation and the proof is complete. \square

We can prove a stronger statement in the specific case where the graphs share algebraic connectivity, λ_1 .

Theorem 3.1.16. *Consider the assumptions given in Theorem 3.1.15 with the added assumption that the common eigenvalue $\lambda > 0$ is the algebraic connectivity, i.e., the lowest nonzero eigenvalue of the graphs G_j . Then λ is an eigenvalue of $G(V, E) = G(\cup_{j=1}^n V_j, \cup_{j=1}^n E_j \cup E_0)$ but not the lowest nonzero eigenvalue. Hence, $\varphi(x)$ as defined in Theorem 3.1.15 is an eigenvector of G but not its Fiedler vector.*

Proof. It suffices to prove the theorem for $n = 2$. Let $G_1(V_1, E_2)$ and $G_2(V_2, E_2)$ have Fiedler vectors $\varphi_{(1)}$ and $\varphi_{(2)}$, respectively. We can decompose each vertex set into its positive, negative, and zero sets, i.e., $V_j = i_+(\varphi_{(j)}) \cup i_-(\varphi_{(j)}) \cup i_0(\varphi_{(j)})$. Furthermore, $i_+(\varphi_{(j)})$ and $i_-(\varphi_{(j)})$ are connected subgraphs of G_j .

Now consider the larger graph $G(V, E)$. The function $\varphi(x) := \varphi_{(j)}(x)$ for $x \in V_j$ is an eigenfunction of G by Theorem 3.1.15. However, now, the sets $i_+(\varphi)$ and $i_-(\varphi)$ are disconnected. Indeed, let $x \in i_+(\varphi_{(1)})$ and $y \in i_+(\varphi_{(2)})$. Then any

path connecting x and y must contain an edge in E_0 since all E_0 contains all edges connecting G_1 to G_2 . And hence any path connect x to y will contain at least two vertices in $i_0(\varphi)$.

Then by [69] since $i_+(\varphi)$ and $i_-(\varphi)$ are both disconnected, then φ cannot be the Fiedler vector of G and λ is not the smallest nonzero eigenvalue. \square

We can state a generalization of Theorem 3.1.15 for eigenvectors supported on subgraphs of G .

Theorem 3.1.17. *Consider the graph $G(V, E)$. Let $S \subseteq V$ and let $H(S, E_S)$ be the resulting subgraph defined by just the vertices of S . Suppose that $\varphi^{(S)}$ is an eigenvector of L_S , the Laplacian of subgraph H , with corresponding eigenvalue λ . If $E(S, V \setminus S) = E(i_0(\varphi^{(S)}), V \setminus S)$, that is, if all edges connecting graph H to its complement have a vertex in the zero-set of $\varphi^{(S)}$, then λ is an eigenvalue of G with eigenvector*

$$\varphi(x) = \begin{cases} \varphi^{(S)}(x) & x \in S \\ 0 & x \notin S. \end{cases}$$

Proof. The proof is similar to the proof of Theorem 3.1.15 in that we will simply verify that $L\varphi(x) = \lambda\varphi(x)$ at every point $x \in V$. For any $x \in \int(S)$, then $L\varphi(x) = L_S\varphi^{(S)}(x) = \lambda\varphi^{(S)}(x) = \lambda\varphi(x)$. For any $x \in \int(V \setminus S)$, then $L\varphi(x) = 0$ since φ vanishes at x and all of its neighbors. For $x \in \delta(S)$ (recall $\delta(S) = \{x \in S : (x, y) \in E \text{ and } y \in V \setminus S\}$), we have

$$\begin{aligned} L\varphi(x) &= \sum_{y \sim x} (\varphi(x) - \varphi(y)) = \sum_{\substack{y \sim x \\ y \in S}} (\varphi(x) - \varphi(y)) + \sum_{\substack{y \sim x \\ y \notin S}} (\varphi(x) - \varphi(y)) \\ &= L_S\varphi^{(S)}(x) + (0 - 0) = \lambda\varphi^{(S)}(x) = \lambda\varphi(x), \end{aligned}$$

where the term $(0 - 0)$ arises from the fact that $\varphi(y) = 0$ since $y \notin S$ and since $(x, y) \in E$ then by assumption $x \in i_0(\varphi^{(S)})$ and hence $\varphi(x) = 0$. The same logic shows that $L\varphi(x) = 0$ for $x \in \delta(V \setminus S)$. Hence, we have shown that $L\varphi(x) = \lambda\varphi(x)$ for every possible vertex $x \in V$ and the proof is complete. \square

Theorem 3.1.17 is interesting because it allows us to obtain eigenvalues and eigenvectors of graphs by inspecting for certain subgraphs. Furthermore since the eigenvector is supported on the subgraph, it is sparse and has a large nodal set.

Example 3.1.18. Consider the star graph $S_N(V, E)$ which is complete bipartite graph between N vertices in one class (V_A) and 1 vertex in the other (V_B). Let S be the subgraph formed by any two vertices in V_A and the one vertex in V_B . Then the resulting subgraph on S is the path graph on 3 vertices, P_3 . It is known that P_3 has Fiedler vector $\varphi^{(S)} = (\sqrt{2}, 0, -\sqrt{2})$ and eigenvalue $\lambda = 1$. Then by Theorem 3.1.17, the star graph S_N has eigenvalue $\lambda = 1$ with eigenvector supported on two vertices. In fact, S_N contains exactly $\binom{N}{2}$ path subgraphs all of which contain the center vertex and have $\varphi^{(S)}$ as an eigenvector. However, only $N - 1$ of them will be linearly independent. This method of recognizing subgraphs explains why S_N has eigenvalue 1 with multiplicity $N - 1$ and we have identified a set of basis vectors for that eigenspace.

Chapter 4

Spectral Graph Wavelets

In this chapter we create a generalization of the already established spectral graph wavelets that we call vertex-dynamic spectral graph wavelets. We prove in Theorem 4.3.1 that this generalization still forms a frame for the space of functions on graphs. We present numerical results and end the chapter that both formulations of spectral graph wavelets to not admit a multiresolution analysis unlike classical wavelets.

4.1 Introduction to the spectral graph wavelet transform

Wavelets on graphs were first introduced in [35] using spectral graph theory and Fourier analysis on graphs derived from [59]. The authors define the spectral graph wavelet transform in terms of a kernel function $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ which acts as a band-pass filter. In particular, it satisfies $g(0) = 0$ and $\lim_{x \rightarrow \infty} g(x) = 0$. One has freedom in choosing the particular kernel g in implementation and some examples are given in [35, Section 8].

With the spectral graph wavelet kernel g fixed, we define the wavelet operator $T_g = g(L)$ which acts on a function f as the following Fourier multiplier,

$$\widehat{T_g f}(k) = g(\lambda_k) \hat{f}(k).$$

Hence, taking the inverse Fourier transform yields

$$(T_g f)(m) = \sum_{k=0}^{N-1} g(\lambda_k) \hat{f}(k) \varphi_k(m).$$

The wavelet operator at scale t is defined by $T_g^t = g(tL)$. Notice that even the spatial/vertex domain is discrete as well as its frequency/spectral domain, i.e., $\sigma(L) = \{\lambda_k\}_{k=0}^{N-1}$. However, since we define the spectral wavelet kernel, g , to be defined on a continuum, we are able to define the wavelet operator at arbitrary positive scales.

We define the spectral graph wavelets by applying the wavelet operator to the indicator on a single vertex, $\mathbb{1}_n$, denoted

$$\psi_{t,n} = T_g^t \mathbb{1}_n.$$

Recalling that $\widehat{\mathbb{1}_n}(k) = \varphi_k^*(n)$, the wavelet can be expanded as

$$\psi_{t,n}(m) = \sum_{k=0}^{N-1} g(t\lambda_k) \varphi_k^*(n) \varphi_k(m). \quad (4.1)$$

The wavelet coefficients of a function $f : V \rightarrow \mathbb{R}$ are then defined to be the inner product of f with these wavelets, denoted by

$$W_f(t, n) = \langle f, \psi_{t,n} \rangle. \quad (4.2)$$

By expanding (4.2) and using the definition of the Fourier transform of the real function f , we can express the wavelet coefficients as

$$W_f(t, n) = (T_g^t f)(n) = \sum_{k=0}^{N-1} g(t\lambda_k) \hat{f}(k) \varphi_k(n), \quad (4.3)$$

where t and n are the scale and vertex/space parameters, respectively.

By construction, we have that the spectral graph wavelets, $\psi_{t,n}$, are all orthogonal to the constant eigenfunction φ_0 . Furthermore, if the kernel g is continuous, then the spectral graph wavelets are nearly orthogonal to φ_k for small values of λ_k . One can see this by calculating

$$\langle \varphi_k, \psi_{t,n} \rangle = \sum_{\ell=0}^{N-1} g(t\lambda_\ell) \hat{\varphi}_k(\lambda_\ell) \varphi_\ell(n) = g(t\lambda_k) \varphi_k(n).$$

So as long as λ_k is sufficiently small (dependent on continuity conditions on g) then $g(t\lambda_k)$, and hence $|\langle \varphi_k, \psi_{t,n} \rangle|$, will be close to $g(0) = 0$.

By Corollary 2.4.11, the wavelet transform T_g is not invertible since for any $t > 0$, $g(t\lambda_0) = g(0) = 0$. It is, however, invertible if we restrict to the class of functions with zero mean. If we hope to reconstruct any function $f : V \rightarrow \mathbb{C}$ from the wavelet decomposition, it is necessary to introduce a second class of waveforms to represent the low-frequency content of the function f , analogous to the lowpass residual scaling functions from classical wavelet analysis. These spectral graph scaling functions are determined by a single function $h : \mathbb{R}^+ \rightarrow \mathbb{R}$ satisfying $h(0) > 0$ and $\lim_{x \rightarrow \infty} h(x) = 0$. Since $h(0) > 0$ (and recall $g(0) = 0$), then h retains low-frequency information from f and acts as a low-pass filter. Just as was done for the spectral graph wavelets, the scaling functions are given by $\phi_n = T_h \mathbb{1}_n = h(L) \mathbb{1}_n$, and the scaling coefficients are denoted by

$$S_f(n) = \langle f, \phi_n \rangle.$$

4.2 Wavelets with univariate generating kernels

Both the spectral graph wavelet kernel, g , and spectral graph scaling function, h , are univariate real functions defined on \mathbb{R}^+ . Furthermore, both the spectral graph wavelet and scaling functions both depend on the continuous scale parameter, t . For any practical computation or numerical implementation, the scale parameter, t , must be sampled into a finite number of scales $\{t_j\}_{j=1}^J$. This will produce a family of NJ wavelets $\psi_{t_j,n}$ and N scaling functions ϕ_n (since the scaling function is independent of the scale parameter).

One of the reasons why wavelet decompositions are widely used in classical analysis is that they efficiently represent functions and form frame. We present the following theorem which states under what conditions the spectral graph wavelets and scaling functions form a frame.

Theorem 4.2.1. *[35, Theorem 5.8] Given a finite sample of scales $\{t_j\}_{j=1}^J$, the set $F = \{\phi_n\}_{n=1}^N \cup \{\psi_{t_j,n}\}_{j=1,n=1}^{J,N}$ forms a frame with bounds A, B given by*

$$A = \min_{\lambda \in [0, \lambda_{N-1}]} G(\lambda),$$

$$B = \max_{\lambda \in [0, \lambda_{N-1}]} G(\lambda),$$

where

$$G(\lambda) = h^2(\lambda) + \sum_{j=1}^J g(t_j \lambda)^2.$$

Proof. Fix a function f . Using the expansion (4.3) and the orthonormality of the

eigenvectors of the Laplacian, we have

$$\begin{aligned}
\sum_{n=1}^N |W_f(t, n)|^2 &= \sum_{n=1}^N \sum_{k=0}^{N-1} g(t\lambda_k) \varphi_k(n) \hat{f}(k) \sum_{k'=0}^{N-1} \overline{g(t\lambda_{k'}) \varphi_{k'}(n) \hat{f}(k')} \\
&= \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} g(t\lambda_k) \overline{g(t\lambda_{k'})} \hat{f}(k) \overline{\hat{f}(k')} \sum_{n=1}^N \varphi_k(n) \overline{\varphi_{k'}(n)} \\
&= \sum_{k=0}^{N-1} \sum_{k'=0}^{N-1} g(t\lambda_k) \overline{g(t\lambda_{k'})} \hat{f}(k) \overline{\hat{f}(k')} \delta(k, k') \\
&= \sum_{k=0}^{N-1} |g(t\lambda_k)|^2 |\hat{f}(k)|^2.
\end{aligned}$$

A similar computation reveals

$$\sum_{n=1}^N |S_f(n)|^2 = \sum_{k=0}^{N-1} |h(\lambda_k)|^2 |\hat{f}(k)|^2.$$

Let Q denote the sum of the squares of inner products of f with each of the elements of F , i.e.,

$$Q = \sum_{j=1}^J \sum_{n=1}^N |\langle f, \psi_{t_j, n} \rangle|^2 + \sum_{n=1}^N |\langle f, \phi_n \rangle|^2.$$

Then the previous computations give

$$Q = \sum_{k=0}^{N-1} \left(|h(\lambda_k)|^2 + \sum_{j=1}^J |g(t_j \lambda_k)|^2 \right) |\hat{f}(k)|^2 = \sum_{k=0}^{N-1} G(\lambda_k) |\hat{f}(\lambda_k)|^2.$$

By definition of A and B above, we have

$$A \sum_{k=0}^{N-1} |\hat{f}(\lambda_k)|^2 \leq Q \leq B \sum_{k=0}^{N-1} |\hat{f}(\lambda_k)|^2.$$

Finally, Parseval's identity $\|f\|^2 = \sum |\hat{f}(k)|^2$ shows the desired frame inequality.

□

In wavelet analysis, it is not enough to merely compute the wavelet coefficients of a signal; often one is required to reconstruct a signal from its collection of wavelet

coefficients. In [35], the authors discuss that the spectral graph wavelet transform is an overcomplete transform. Among the wavelets $\psi_{t_j, n}$, and scaling functions, ϕ_n , the wavelet frame consists of $N(J + 1)$ coefficients (where J is the number of scaling parameters sampled). If we write the coefficients as the $N(J + 1)$ vector $c = Wf$ then it is well known that W has an infinite number of left-inverses M so that $MWf = f$. However, the authors discuss that a natural choice among these inverses is the pseudoinverse $M = (W^*W)^{-1}W^*$. The pseudoinverse satisfies the minimum-norm property

$$Mc = \arg \min_{f \in \mathbb{R}^N} \|c - Wf\|_2.$$

4.3 Vertex-dynamic spectral graph wavelets

We now extend the theory from [35] to create spectral graph wavelets whose wavelet kernel g not only depends on scale and frequency, but also vertex location. The spectral graph wavelet transform (4.2) depends on a scale parameter, t , and a translation parameter, n . However, in the expansion (4.3), the wavelet kernel term $g(t\lambda_k)$ does not depend on the translation parameter, n . We extend the theory to define wavelets using a bivariate wavelet kernel $\mathbf{g}(n, t\lambda_k)$ which both depends on scale and translation parameters. This will provide the user more freedom when performing wavelet analysis on data graphs. The user has the freedom to widen the support of wavelets at regions where higher redundancy is desired or, conversely, to narrow the support of the kernel at regions where overrepresentation of a function is less necessary. Because of this added vertex-dependency of the wavelet kernel,

we shall call the wavelets developed in this subsection *vertex-dynamic spectral graph wavelets*.

Following the setup in (4.1), we shall define the vertex-dynamic spectral graph wavelets by

$$\boldsymbol{\psi}_{t,n}(m) = \sum_{k=0}^{N-1} \mathbf{g}(n, t\lambda_k) \varphi_k^*(n) \varphi_k(m),$$

the only difference now, being the fact that $\mathbf{g} : V \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is bivariate. We use the bold-face notation to denote the bivariate spectral graph wavelet kernel and the wavelet functions it generates. We still impose that \mathbf{g} act as a high-pass filter, i.e., the kernel satisfies $\mathbf{g}(n, 0) = 0$ for all n and $\lim_{t \rightarrow \infty} \mathbf{g}(n, t) = 0$ for all n .

Similarly to what was done in Section 4.1, we can define a vertex-dynamic scaling function

$$\boldsymbol{\phi}_n(m) = \sum_{k=0}^{N-1} \mathbf{h}(n, \lambda_k) \varphi_k^*(n) \varphi_k(m), \quad (4.4)$$

where the scaling function $\mathbf{h} : V \times \mathbb{R}^+ \rightarrow \mathbb{R}$ is now bivariate. It still acts as a low-pass filter and satisfies $\mathbf{h}(n, 0) > 0$ for all n and $\lim_{t \rightarrow \infty} \mathbf{h}(n, t) = 0$ for all n .

With the wavelet and scaling functions, we define the vertex-dynamic wavelet and scaling coefficients by

$$W_f(t, n) = \langle f, \boldsymbol{\psi}_{t,n} \rangle, \quad S_f(n) = \langle f, \boldsymbol{\phi}_n \rangle.$$

Theorem 4.3.1. *Given a finite sample of scales $\{t_j\}_{j=1}^J$, the set $F = \{\boldsymbol{\phi}_n\}_{n=1}^N \cup \{\boldsymbol{\psi}_{t_j,n}\}_{j=1,n=1}^{J,N}$ forms a frame with bounds A, B given by*

$$A = \left(\min_{\substack{n=1,\dots,N \\ k=0,\dots,N-1}} h(n, \lambda_k)^2 + \sum_{j=1}^J \min_{\substack{n=1,\dots,N \\ k=0,\dots,N-1}} g(n, t_j \lambda_k)^2 \right),$$

$$B = \left(\max_{\substack{n=1,\dots,N \\ k=0,\dots,N-1}} h(n, \lambda_k)^2 + \sum_{j=1}^J \max_{\substack{n=1,\dots,N \\ k=0,\dots,N-1}} g(n, t_j \lambda_k)^2 \right).$$

Proof. For a given scale $t \in \mathbb{R}^+$, we can compute

$$\begin{aligned} \sum_{n=1}^N |W_f(t, n)|^2 &= \sum_{n=1}^N \sum_{k=0}^{N-1} \hat{f}(k) \mathbf{g}(n, t\lambda_k) \varphi_k(n) \overline{\sum_{k'=0}^{N-1} \hat{f}(k') \mathbf{g}(n, t\lambda_{k'}) \varphi_{k'}(n)} \\ &= \left\| \sum_{k=0}^{N-1} \hat{f}(k) \mathbf{g}(\cdot, t\lambda_k) \varphi_k(\cdot) \right\|^2 \\ &\geq \min_{\substack{n=1,\dots,N \\ k=0,\dots,N-1}} \mathbf{g}(n, t\lambda_k)^2 \left\| \sum_{k=0}^{N-1} \hat{f}(k) \varphi_k(\cdot) \right\|^2 \\ &= \min_{\substack{n=1,\dots,N \\ k=0,\dots,N-1}} \mathbf{g}(n, t\lambda_k)^2 \|f\|^2, \end{aligned}$$

where the last inequality follows from the definition of the inverse Fourier transform.

Similarly we can compute

$$\sum_{n=1}^N |W_f(t, n)|^2 \leq \max_{\substack{n=1,\dots,N \\ k=0,\dots,N-1}} \mathbf{g}(n, t\lambda_k)^2 \|f\|^2,$$

and

$$\min_{\substack{n=1,\dots,N \\ k=0,\dots,N-1}} \mathbf{h}(n, \lambda_k)^2 \|f\|^2 \leq \sum_{n=1}^N |S_f(n)|^2 \leq \max_{\substack{n=1,\dots,N \\ k=0,\dots,N-1}} \mathbf{h}(n, \lambda_k)^2 \|f\|^2.$$

Let Q denote the sum of the squares of inner products of f with each of the elements of F , i.e.,

$$Q = \sum_{j=1}^J \sum_{n=1}^N |\langle f, \psi_{t_j, n} \rangle|^2 + \sum_{n=1}^N |\langle f, \phi_n \rangle|^2.$$

Then we have the estimate

$$A \|f\|^2 \leq Q \leq B \|f\|^2$$

as desired. \square

Corollary 4.3.2. *Consider a finite sample of scales $\{t_j\}_{j=1}^J \subseteq \mathbb{R}^+$ listed in increasing order. Suppose $0 < m_h \leq h(\cdot, \lambda) \leq M_h$ on $\lambda \in [0, \lambda_1]$ and $0 < m_g \leq g(\cdot, \lambda) \leq M_g$ for all $\lambda \in [t_1\lambda_1, t_J\lambda_{N-1}]$. Then the set $F = \{\phi_n\}_{n=1}^N \cup \{\psi_{t_j, n}\}_{j=1, n=1}^{J, N}$ forms a frame with frame inequality*

$$(m_h^2 + Jm_g^2) \|f\|^2 \leq Q \leq (M_h^2 + JM_g^2) \|f\|,$$

where Q denotes the sum of the squares of inner products of f with each of the elements of F , i.e.,

$$Q = \sum_{j=1}^J \sum_{n=1}^N |\langle f, \psi_{t_j, n} \rangle|^2 + \sum_{n=1}^N |\langle f, \phi_n \rangle|^2.$$

4.4 Numerical Implementation

We devote this subsection to illustrating numerical implementations of the spectral graph wavelets. In all of the following figures, wavelets were created for the Minnesota graph from [23] which is a graph of 2640 vertices and 3302 edges.

The spectral graph wavelet kernel used is the one proposed in Section 8.1 of [35],

$$g(\lambda) = \begin{cases} x_1^{-\alpha} \lambda^\alpha & \text{for } \lambda < x_1 \\ s(\lambda) & \text{for } x_1 \leq \lambda \leq x_2 \\ x_2^\beta \lambda^{-\beta} & \text{for } \lambda > x_2. \end{cases} \quad (4.5)$$

One can choose the positive parameters x_1, x_2, α, β and any smooth $s(\lambda)$ that makes g continuous. For our numerical simulations, we chose $x_1 = 1, x_2 = 2, \alpha = \beta = 1$ and $s(\lambda) = -5 + 11\lambda - 6\lambda^2 + \lambda^3$. Our kernel, g , is shown in Figure 4.1.

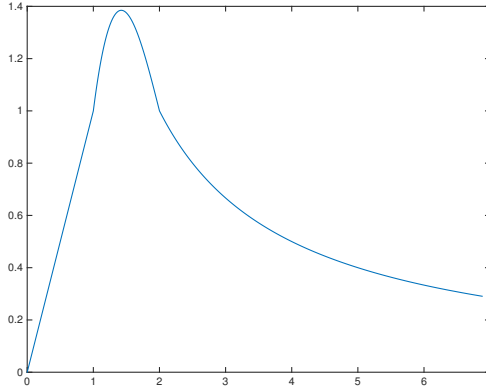


Figure 4.1: One choice of the spectral graph wavelet kernel, $g(\lambda)$.

Figure 4.2 shows plots of the wavelet functions $\psi_{t,n}$ for three different choices of vertices, n , and four scales, t .

We also generate examples of spectral graph wavelets with bivariate wavelet kernels. Both Figures 4.5 and 4.6 show twelve wavelets $\psi_{t,n}$ on the Minnesota graph just as was done in Figure 4.2, except they are generated from a bivariate wavelet kernel $\mathbf{g}(n, t)$.

First, Figure 4.5 shows wavelets using the kernel $\mathbf{g}_1(n, t) = g(ty(n))$ (see Figure 4.3), where $y(n)$ is the vertical coordinate of the n 'th node in the given data map and $g(\cdot)$ is the univariate wavelet kernel from (4.5). With this choice of \mathbf{g}_1 , we effectively scale the kernel for higher northern modes more than we do for more southern nodes. This causes northern modes to have a larger support than those wavelets shown in Figure 4.2 while the southern-most nodes stay well-localized.

Secondly, Figure 4.6 shows wavelets using the kernel $\mathbf{g}_2(n, t) = g(tR(1786, n))$ (see Figure 4.3), where recall from Definition 1.2.2 that $R(1786, n)$ is the effective

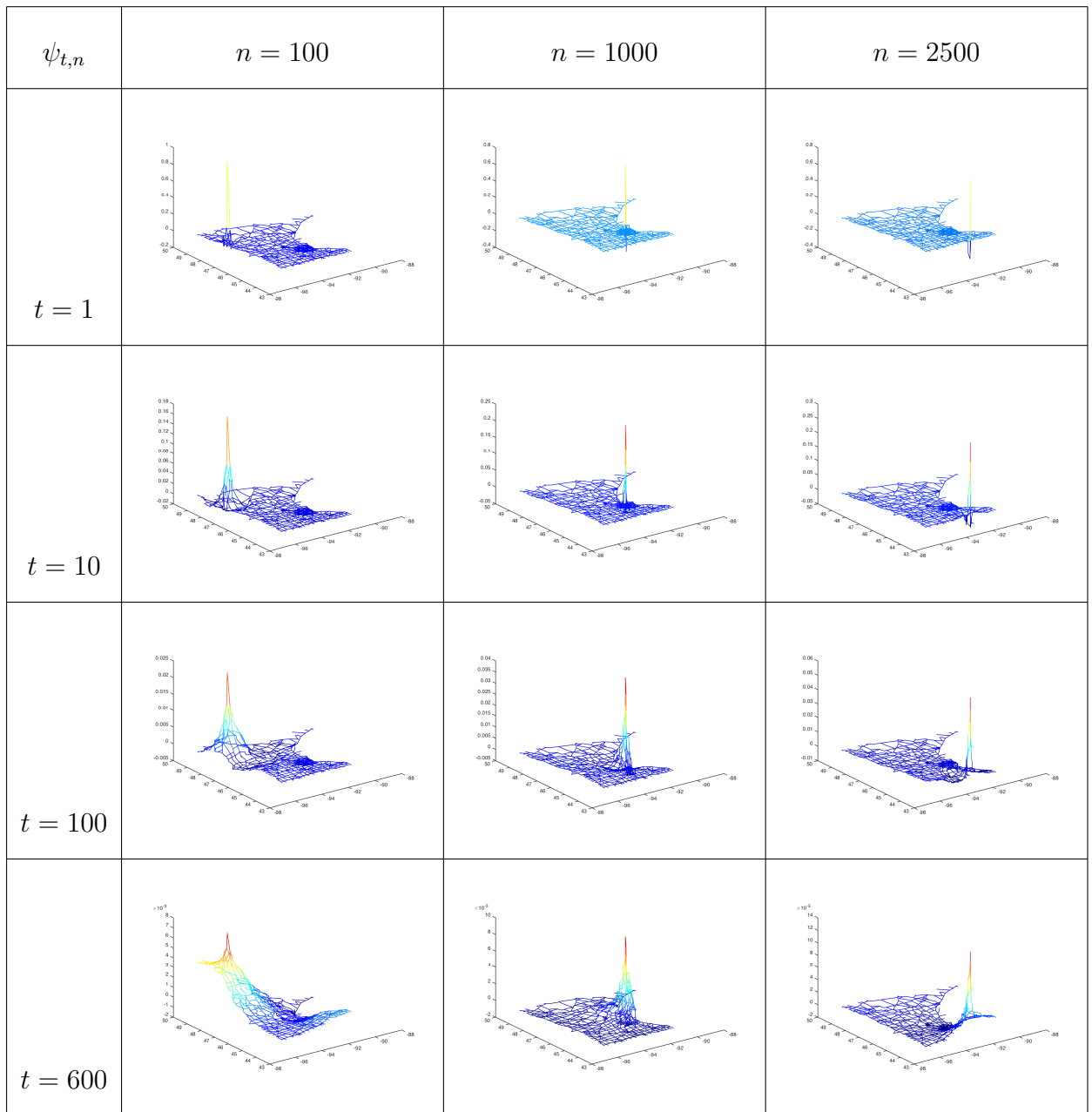


Figure 4.2: Twelve wavelet functions $\psi_{t,n}$ on the Minnesota graph at three different vertices $n = 100, 1000, 2500$ and four different scales $t = 1, 10, 100, 600$.

resistance between node n and node 1786. The node 1786 was chosen because it is the vertex that minimizes the sum of effective resistances among all other nodes, that is, $1786 = \arg \min_j \sum_{i=1}^N R(i, j)$. It is the vertex with the smallest average resistance among all other nodes. This gives a notion of “centrality” on the graph and it does experimentally coincide with a vertex near the geometric center of the graph, see Figure 4.4.

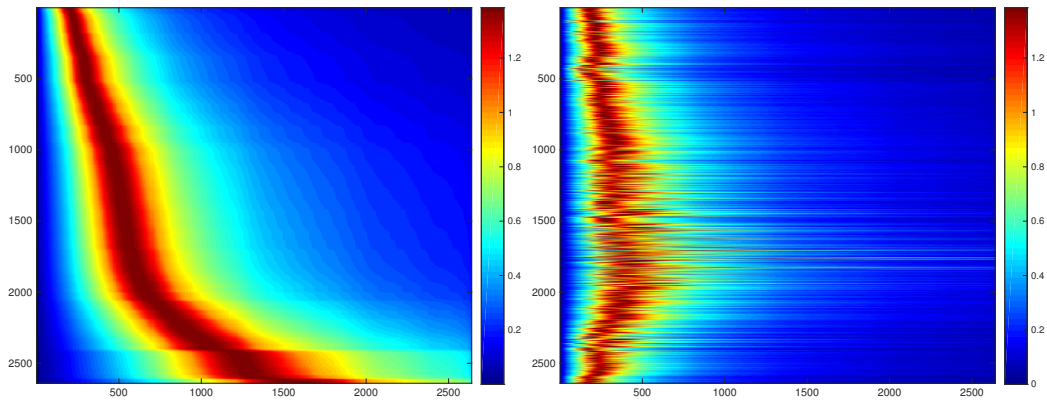


Figure 4.3: Plot of the bivariate kernel functions $\mathbf{g}_1(n, t) = g(ty(n))$ (left) and $\mathbf{g}_2(n, t) = g(tR(1786, n))$ (right) used to generate the wavelets shown in Figures 4.5 and 4.6, respectively. Here, $y(n)$ denotes the vertical coordinate of the n 'th node in the date graph, and R represents the effective resistance between nodes. Each row in the above image corresponds to a differently scaled version of the univariate kernel function g given in (4.5).

As is illustrated by Figure 4.6 the wavelets centered near vertex 1786 (i.e. $\psi_{t,1000}$) have support and behavior very similar to the univariate wavelets shown in Figure 4.2. However, the wavelets further from the center of the graph (i.e., $\psi_{t,100}$

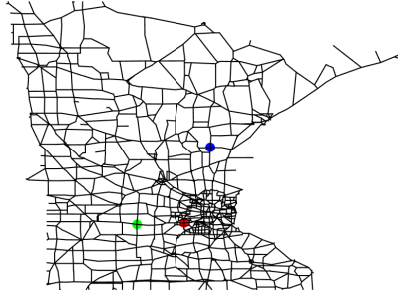


Figure 4.4: Three notions of centrality using the effective resistance on a graph. For the Minnesota graph we mark $x_1 = \arg \min_j \sum_{i=1}^N R(i, j)$ (red) and $x_2 = \arg \min_j \sum_{i=1}^N R(i, j)^2$ (green). Furthermore, we mark $x_0 = \arg \min_j \max_i R(i, j)$ (blue) which has the shortest longest path in terms of effective resistance.

and $\psi_{t,2500}$) have much wider support than the corresponding univariate wavelets shown in Figure 4.2.

4.5 Lack of Multiresolution Analysis

We devote this subsection to present that the graph wavelets presented in this section do not exhibit a multiresolution structure that makes wavelets in Euclidean space such a versatile tool.

First let us recall the definition of a multiresolution analysis for the simple case of $L^2(\mathbb{R})$.

Definition 4.5.1 ([66]). A *multiresolution analysis* (MRA) is an increasing se-

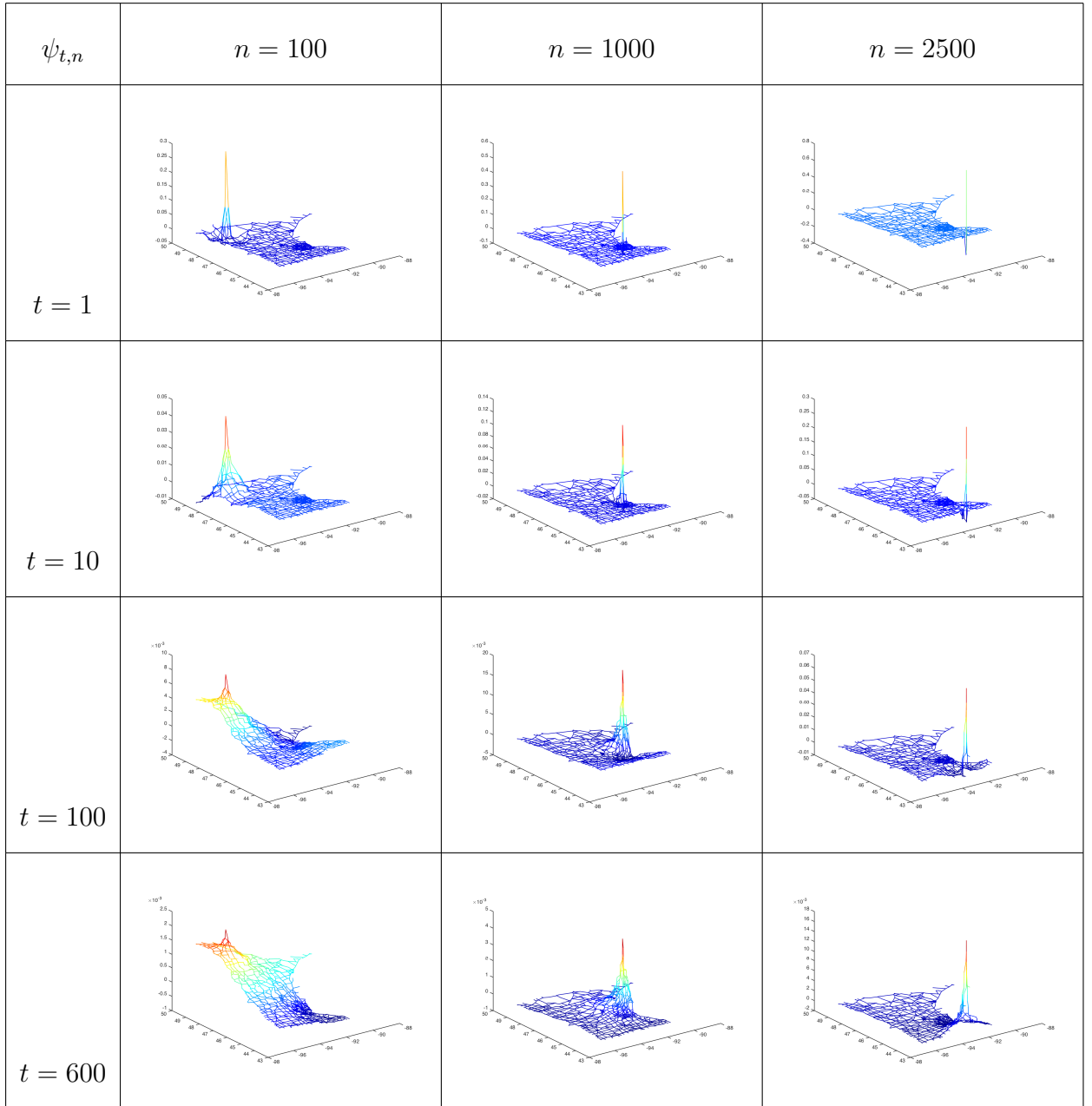


Figure 4.5: Twelve wavelet functions $\psi_{t,n}$ on the Minnesota graph at three different vertices $n = 100, 1000, 2500$ and four different scales $t = 1, 10, 100, 600$ using the bivariate spectral graph wavelet kernel $\mathbf{g}_1(n, t) = g(ty(n))$, where g is the univariate kernel given in (4.5) and $y(n)$ is the vertical coordinate of the n 'th node in the map.

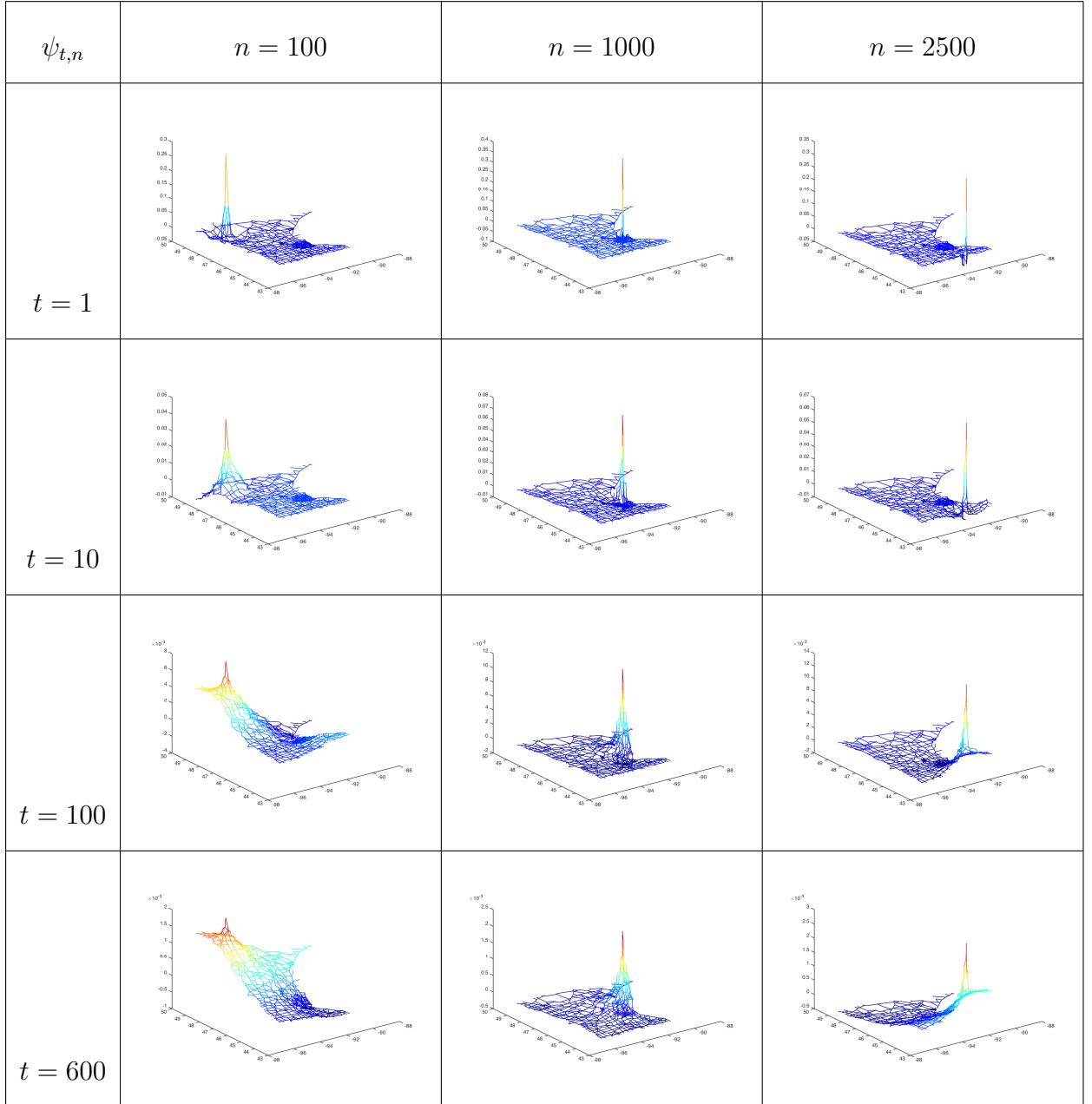


Figure 4.6: Twelve wavelet functions $\psi_{t,n}$ on the Minnesota graph at three different vertices $n = 100, 1000, 2500$ and four different scales $t = 1, 10, 100, 600$ using the bivariate spectral graph wavelet kernel $\mathbf{g}_2(n, t) = g(tR(1786, n))$, where g is the univariate kernel given in (4.5) and R is the effective resistance between vertices.

quence of subspaces of $L^2(\mathbb{R})$, $\cdots \subseteq V_{-1} \subseteq V_0 \subseteq V_1 \subseteq \cdots$, with a scaling function φ that satisfy:

- (i) (Density) $\cup_j V_j$ is dense in $L^2(\mathbb{R})$,
- (ii) (Separation) $\cap_j V_j = \{0\}$,
- (iii) (Scaling) $f(x) \in V_j$ iff $f(2^{-j}x) \in V_0$,
- (iv) (Orthonormality) $\{\varphi(x - \gamma)\}_{\gamma \in \mathbb{Z}}$ is an orthonormal basis for V_0 .

The classic example of a multiresolution analysis on $L^2(\mathbb{R})$ uses the indicator function $\varphi(x) = \mathbb{1}_{[0,1]}(x)$. Let V_0 denote the linear span of integer translates of φ . Then, $\{\varphi(x - \gamma)\}_{\gamma \in \mathbb{Z}}$ forms an orthonormal basis for V_0 , which is the space of all square-integrable functions that are piecewise constant on intervals with integer endpoints. That $\{\varphi(x - \gamma)\}_{\gamma \in \mathbb{Z}}$ is an orthonormal set can easily be verified since each $\varphi(\cdot - \gamma)$ is supported on a distinct interval in \mathbb{R} . Further, one can observe that $\varphi(x - \gamma) = \varphi(2x - 2\gamma) + \psi(2x - 2\gamma - 1)$ which gives the relation $V_0 \subseteq V_1$ and the scaling property of Definition 4.5.1. In general, simply by dilating by factors of 2^{-j} construct the other function spaces V_j . The Haar system (i.e., functions that are piecewise constant on $\cup_j V_j$) is dense in the space of simple functions and by [7, Theorem 5.5.3] simple functions are dense in $L^2(\mathbb{R})$ which proves the density property from Definition 4.5.1. Finally, it is elementary to show that only the zero function is square-integrable and piecewise constant on dyadic intervals $[2^{-j}k, 2^{-j}(k + 1)]$ for any $j \in \mathbb{Z}$, which gives the separation property of Definition 4.5.1. Hence, we have

shown that the function spaces $\{V_j\}$ and the scaling function φ form an MRA for $L^2(\mathbb{R})$.

However, while we have an orthonormal basis for each V_j , we cannot simply take the union of these bases as an orthonormal basis for all of $L^2(\mathbb{R})$. Indeed, although $V_j \subseteq V_{j+1}$ the orthonormal basis $\{2^{j/2}\varphi(2^j x - \gamma)\}$ for V_j is not contained in the orthonormal basis $\{2^{(j+1)/2}\varphi(2^{j+1}x - k)\}$ for V_{j+1} . We remedy this by introducing the orthogonal complement of V_0 in V_1 , call it W_0 . We write in terms of the Hilbert space direct sum $V_1 = V_0 \oplus W_0$. We wish to obtain a function, ψ , whose integer translates form an orthonormal basis for the function space W_0 . Enter the Haar wavelet, $\psi(x)$, defined by

$$\psi(x) = \begin{cases} 1 & \text{if } 0 \leq x < 1/2 \\ -1 & \text{if } 1/2 \leq x < 1 \\ 0 & \text{otherwise.} \end{cases}$$

The Haar wavelet is contained in the space V_1 since it is piecewise constant on intervals of length $1/2$ and it is orthogonal to all of V_0 . One can also verify that the integer translates of the Haar wavelet forms an orthonormal basis for W_0 . Now, we can dilate *both* V_j and W_j to obtain the relation $V_{j+1} = V_j \oplus W_j$. Combining these results gives $L^2(\mathbb{R}) = \bigoplus_{j=-\infty}^{\infty} W_j$ and since each of the spaces W_j are all mutually orthogonal, we now have an orthonormal basis $\{2^{j/2}\psi(2^j x - \gamma)\}_{j \in \mathbb{Z}, \gamma \in \mathbb{Z}}$ for $L^2(\mathbb{R})$, known as the Haar wavelet basis. To summarize, once we obtained a multiresolution analysis for with a scaling function, φ , we were then able to take orthogonal complements to obtain a mother wavelet, whose translations and dilations form an orthonormal basis for the whole function space.

We will demonstrate that the spectral graph wavelets developed in this section do not exhibit the nice multiresolution analysis on the real line.

We have defined the scaling function, ϕ_n , on graphs by (4.4), which is defined in terms of a low-pass kernel $\mathbf{h}(n, t)$. The translates of the graph scaling function $\{\phi_n\}_{n \in V}$ does not form an orthogonal family. Indeed, for any two distinct vertices $n, p \in V$, then

$$\begin{aligned}
\langle \phi_n, \phi_p \rangle &= \sum_{m=1}^N \sum_{k=0}^{N-1} \mathbf{h}(n, \lambda_k) \varphi_k^*(n) \varphi_k(m) \sum_{\ell=0}^{N-1} \mathbf{h}(p, \lambda_\ell) \varphi_\ell(p) \varphi_\ell^*(m) \\
&= \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} \mathbf{h}(n, \lambda_k) \mathbf{h}(p, \lambda_\ell) \varphi_k^*(n) \varphi_\ell(p) \sum_{m=1}^N \varphi_k(m) \varphi_\ell^*(m) \\
&= \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} \mathbf{h}(n, \lambda_k) \mathbf{h}(p, \lambda_\ell) \varphi_k^*(n) \varphi_\ell(p) \delta(k, \ell) \\
&= \sum_{k=0}^{N-1} \mathbf{h}(n, \lambda_k) \mathbf{h}(p, \lambda_k) \varphi_k^*(n) \varphi_k(p). \tag{4.6}
\end{aligned}$$

We can assert that $\sum_{k=0}^{N-1} \varphi_k^*(n) \varphi_k(p) = 0$ for $n \neq p$ since $\Phi \Phi^* = \Phi^* \Phi = I_N$. However by including the positive factors $\mathbf{h}(n, \lambda_k) \mathbf{h}(p, \lambda_k)$, we cannot ensure that $\langle \phi_n, \phi_m \rangle = 0$. In fact, we prove that the univariate graph scaling function will not form an orthogonal family in Theorem 4.5.4.

Lemma 4.5.2. *The Hermitian matrix $A = [a_{i,j}]_{i,j=1}^N$ is diagonal if and only if its eigenvector matrix is a row permutation of the identity matrix.*

Proof. One can easily compute that for any $i = 1, \dots, N$, we have $Ae_i = a_{i,i} \cdot e_i$ which proves that the sufficiency claim to the lemma. In this case, the eigenvalues of A are precisely the entries along its diagonal.

The necessary condition also follows simply. Since A is Hermitian, the spectral theorem guarantees that $A = VDV^*$ for a unitary matrix V and a diagonal D . Our

assumption is that V is some row permutation of the identity matrix. The spectral theorem is invariant to row permutations and so without loss of generality, we may assume that V equals the identity which gives $A = D$ which proves the lemma. \square

Lemma 4.5.3. *The undirected graph G can have Laplacian eigenvector matrix Φ equal to a row permutation of the identity matrix if and only if G has no edges between distinct vertices. If G is simple, then this is precisely the empty graph.*

Proof. Let L denote the Laplacian matrix of G and let e_i denote the indicator vector with a 1 in the i 'th coordinate and zeros elsewhere. Then for any $i = 1, \dots, N$, we can compute $Le_i = [L_{i,1}, L_{i,2}, \dots, L_{i,N}]^\top$, which equals a multiple of e_i if and only if $L_{i,j} = \lambda_i \delta(i, j)$ for some real λ_i and δ is the Kronecker delta. We have $L_{i,j} = \delta(i, j)$ if and only if the vertex i has either no edges or only self loops. In either case the eigenvalue associated to e_i , equals the degree d_i , which is zero for simple graphs. Thus we have shown that $\{e_i\}_{i=1}^N$ is the eigenbasis of L which makes Φ some permutation of the identity, if and only if G has at most self-loops or is empty. \square

Theorem 4.5.4. *The graph wavelet scaling function ϕ will be orthogonal among its translations if and only if the graph is empty or contains only self-loops.*

Proof. By repeating the calculation in (4.6), but replacing the bivariate scaling kernel $\mathbf{h}(\cdot, \lambda)$ with its univariate counterpart $h(\lambda)$, we get

$$\langle \phi_n, \phi_p \rangle = \sum_{k=0}^{N-1} h(\lambda_k)^2 \varphi_k^*(n) \varphi_k(p),$$

which we can write as

$$\langle \phi_n, \phi_p \rangle = \Phi H \Phi^*,$$

where H is the diagonal matrix with entries $h(\lambda_k)^2$ along the diagonal. To have the scaling function ϕ orthogonal among its translations is equivalent to the matrix $\Phi H \Phi^*$ being diagonal. In other words, Φ , the eigenvector matrix of the graph Laplacian L , must also be the eigenvector matrix of $\Phi H \Phi^*$ which is not necessarily equal to L . By Lemma 4.5.2, $\Phi H \Phi^*$ is diagonal if and only if Φ is the identity matrix (up to permutation). By Lemma 4.5.3, this is possible if and only if G has no edges between vertices. \square

In the real case, the wavelet spaces W_j were all mutually orthogonal. This cannot happen for the spectral graph wavelets; that is, there are no scalings, $t, s \in (0, \infty)$ so that $\langle \boldsymbol{\psi}_{t,n}, \boldsymbol{\psi}_{s,n} \rangle = 0$ since

$$\begin{aligned}
\langle \boldsymbol{\psi}_{t,n}, \boldsymbol{\psi}_{s,n} \rangle &= \sum_{m=1}^N \sum_{k=0}^{N-1} \mathbf{g}(n, t\lambda_k) \varphi_k^*(n) \varphi_k(m) \sum_{\ell=0}^{N-1} \mathbf{g}(n, s\lambda_\ell) \varphi_\ell(n) \varphi_\ell^*(m) \\
&= \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} \mathbf{g}(n, t\lambda_k) \mathbf{g}(n, s\lambda_\ell) \varphi_k^*(n) \varphi_\ell(n) \sum_{m=1}^N \varphi_k(m) \varphi_\ell^*(m) \\
&= \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} \mathbf{g}(n, t\lambda_k) \mathbf{g}(n, s\lambda_\ell) \varphi_k^*(n) \varphi_\ell(n) \delta(k, \ell) \\
&= \sum_{k=0}^{N-1} \mathbf{g}(n, t\lambda_k) \mathbf{g}(n, s\lambda_k) |\varphi_k(n)|^2 > 0,
\end{aligned}$$

since \mathbf{g} is defined to be positive on all of $V \times \mathbb{R}^+$. One can get $\langle \boldsymbol{\psi}_{t,n}, \boldsymbol{\psi}_{s,n} \rangle$ arbitrarily close to zero by sending scales t and s to either 0 or ∞ (since $\mathbf{g}(n, \cdot)$ vanishes in those limits) but there exists no nontrivial orthogonal scaling as exhibited in the classical Euclidean wavelets.

We note that the authors of [33] and [58] attempt to define an MRA for graphs. However, they only do so for a very particular family of tree graphs and do not use any of the notions of translation and dilation as introduced in this section motivated

by spectral graph theory.

We conclude this section by remarking that for the same reasons that the spectral graph wavelets do not admit a multiresolution analysis, they likewise do not admit the more generalized frame multiresolution analysis introduced in [8]. Frame multiresolution analysis (FMRA) is equivalent to the multiresolution analysis definition in Definition 4.5.1 except the orthonormality condition is replaced with the condition that the collection of functions form a frame. The scaling condition in Definition 4.5.1 failing to hold prohibits these spectral graph wavelets from satisfying the conditions to be a FMRA.

Chapter 5

Spectral Graph Sparsification

This chapter presents Spielman's twice Ramanujan graph sparsification results and proves in Theorem 5.1.3 that the algorithm that he presents cannot exceed sparsification that is conjectured to be optimal.

Daniel Spielman and his collaborators have taken on the task of sparsifying large graphs to efficiently speed up large computations on graphs. The sparsification involves deleting edges of a graph in such a way to best preserve the spectral structure of the graph's Laplacian. The three, now-famous, Spielman-Teng papers that studying this problem of sparsifying and solving linear systems in nearly-linear time are [62–64]. These papers and the sparsification results that follow have ignited a push for near-linear sparsification solvers [46, 47, 70]

In [61], Spielman and Srivastava present an algorithm that uses graph effective resistances to sparsify probabilistic algorithm that will satisfactorily sparsify a graph with probability at least $1/2$. One would prefer a deterministic rather than probabilistic algorithm to efficiently sparsify a graph and finally Spielman, Batson, and Srivastava provide such an algorithm producing Twice-Ramanujan sparsifiers in [4]. These results turned out to be more than just a result in graph theory; the methods used in [4] were used in [49] and [50] to prove the then Kadison-Singer conjecture, now known as the Marcus-Spielman-Srivastava theorem; see also [51] and

see [11,14] for connections between the Kadison-Singer problem and frame theory.

5.1 Optimality of Spielman's Twice-Ramanujan Sparsification

In [4], Spielman et al. present an algorithm that sparsifies a weighted finite graph, G , while preserving the spectral properties of the graph Laplacian, L_G . Given a weighted graph $G = (V, E, \omega)$, we say that H is a κ -(spectral) approximation of G if there exist constants $0 < a \leq b$ with $b/a = \kappa$ such that for all $x \in \mathbb{R}^V$, we have

$$a \cdot x^\top L_G x \leq x^\top L_H x \leq b \cdot x^\top L_G x,$$

which we write as

$$aL_G \preceq L_H \preceq bL_G.$$

That is, $A \preceq B$ means that $B - A$ is a positive semidefinite matrix.

The main result of [4] is the following:

Theorem 5.1.1 ([4, Theorem 3.1]). *For every $\epsilon \in (0, 1)$, every undirected weighted graph $G = (V, E, \omega)$ on $|V| = n$ vertices contains a weighted subgraph, $H = (V, F, \tilde{\omega})$, with $\lceil (n-1)/\epsilon^2 \rceil$ edges that satisfies*

$$(1 - \epsilon)^2 L_G \preceq L_H \preceq (1 + \epsilon)^2 L_G,$$

where L_G, L_H are the unnormalized Laplacian matrices for G and H , respectively.

Theorem 5.1.1 can be proved with a change of bases and the following theorem:

Theorem 5.1.2. *Suppose $d > 1$ and v_1, \dots, v_m are vectors in \mathbb{R}^n with*

$$\sum_{i=1}^m v_i v_i^\top = I,$$

then there exist scalars $s_i \geq 0$ with $|\{i : s_i \neq 0\}| \leq \lceil dn \rceil$ such that

$$\left(1 - \frac{1}{\sqrt{d}}\right)^2 I \preceq \sum_{i=1}^m s_i v_i v_i^\top \preceq \left(1 + \frac{1}{\sqrt{d}}\right)^2 I. \quad (5.1)$$

Spielman et al. simultaneously prove Theorem 5.1.2 and present the sparsification algorithm. The algorithm is iterative and starts by setting each scalar $s_i = 0$ and at each iteration, the algorithm chooses one s_i and replaces it with a nonzero quantity. This process is repeated $\lceil dn \rceil$ times. It is shown in [4] that so long as there exist positive constants, $\delta_U, \delta_L, \epsilon_U, \epsilon_L$ satisfying

$$0 \leq 1/\delta_U + \epsilon_U \leq 1/\delta_L - \epsilon_L, \quad (5.2)$$

then after $\lceil dn \rceil$ iterations, the algorithm will always produce a matrix $A = \sum_{i=1}^m s_i v_i v_i^\top$ satisfying

$$\lambda_{\max}(A) \leq n/\epsilon_U + dn\delta_U, \quad \lambda_{\min}(A) \geq -n/\epsilon_L + dn\delta_L.$$

This is significant because then the matrix produced, A , will be a κ -approximation of the identity for

$$\kappa \geq \frac{n/\epsilon_U + dn\delta_U}{-n/\epsilon_L + dn\delta_L} = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}, \quad (5.3)$$

where $\lambda_{\max}(A)$ is the largest eigenvalue of A and $\lambda_{\min}(A)$ denotes the smallest nonzero eigenvalue of A . Spielman et. al. conclude the the proof by choosing

$$\delta_L = 1, \quad \delta_U = \frac{\sqrt{d}-1}{\sqrt{d}+1}, \quad \epsilon_L = \frac{1}{\sqrt{d}}, \quad \epsilon_U = \frac{\sqrt{d}-1}{d+\sqrt{d}},$$

and hence, with this choice of constants, the approximation condition number equals

$$\kappa = \frac{d+2\sqrt{d}+1}{d-2\sqrt{d}+1} = \left(\frac{1+1/\sqrt{d}}{1-1/\sqrt{d}}\right)^2 =: \kappa_d. \quad (5.4)$$

While we don't necessarily have

$$\left(1 - \frac{1}{\sqrt{d}}\right)^2 \leq \lambda_{\min}(A) \leq \lambda_{\max}(A) \leq \left(1 + \frac{1}{\sqrt{d}}\right)^2, \quad (5.5)$$

multiplying A by the appropriate scalar will achieve this distribution of eigenvalues.

Without loss of generality, we assume A to be properly normalized so that (5.5) holds.

We will represent the sparsifying algorithm in [4] by `RamanujanSparsify`. Given a graph $G = (V, E, \omega)$ with Laplacian L_G , and admissible scalars $d, \kappa > 1$, the algorithm will produce a sparsified graph, $H = (V, \tilde{E}, \tilde{\omega})$, with Laplacian $L_H = \text{RamanujanSparsify}(L_G, d, \kappa)$ and at most $\lceil dn \rceil$ edges. A natural question is, given the linear scalar $d > 1$, what is the smallest κ that for which `RamanujanSparsify` will yield a graph H that is a κ -approximation of G ? The answer, amazingly enough, is precisely the Ramanujan constant κ_d . The proof is elementary, yet rather tedious.

Theorem 5.1.3. *Given a graph $G = (V, E, \omega)$ with $|V| = n$ and associated Laplacian matrix L_G , and scalar $d > 1$, then the algorithm `RamanujanSparsify`(L_G, d, κ) will yield the Laplacian matrix L_H for a graph H with at most $\lceil dn \rceil$ edges that is a κ -approximation of G provided that*

$$\kappa \geq \left(\frac{1 + 1/\sqrt{d}}{1 - 1/\sqrt{d}}\right)^2.$$

Proof. In order for `RamanujanSparsify` to run successfully, one just needs to select positive constants, $\delta_U, \delta_L, \epsilon_U, \epsilon_L$ that simultaneously satisfy (5.2) and (5.3). Notice that if $\delta_U, \delta_L, \epsilon_U, \epsilon_L$ simultaneously satisfy (5.2) and (5.3), then $c\delta_U, c\delta_L, \epsilon_U/c, \epsilon_L/c$ will also satisfy (5.2) and (5.3) for any $c > 0$. Hence, without loss of generality, we can select $\delta_U = 1$.

After simplifying the common factor n in (5.3) we have to find positive solutions to the following inequality,

$$\frac{1/\epsilon_U + d}{-1/\epsilon_L + d\delta_L} \leq \kappa.$$

Without loss of generality, one can choose ϵ_U so that the right-hand side of (5.2) holds with equality. Substituting this gives the above inequality in terms of only two variables, ϵ_L and δ_L , and we to solve

$$\frac{d(1 - \epsilon_L\delta_L) - (d - 1)\delta_L}{(1 - \epsilon_L\delta_L - \delta_L)(-1/\epsilon_L + d\delta_L)} \leq \kappa,$$

over the domain $\epsilon_L \in (0, \infty), \delta_L \in (0, (1 + \epsilon_L)^{-1})$. We have to restrict the domain of δ_L to satisfy (5.2).

We can solve the above expression with equality for δ_L (in terms of ϵ_L) and obtain

$$\delta_L = \frac{\kappa - \epsilon_L + d\epsilon_L + (d + 1)\kappa\epsilon_L + d\epsilon_L^2 \pm \sqrt{h_{d,\kappa}(\epsilon_L)}}{2d\kappa\epsilon_L(1 + \epsilon_L)}, \quad (5.6)$$

where $h_{d,\kappa}(\epsilon_L)$ is the discriminant,

$$h_{d,\kappa}(\epsilon_L) = (\epsilon_L(d\epsilon_L + d - 1) + \kappa(1 + (d + 1)\epsilon_L))^2 - 4d\kappa\epsilon_L(1 + \epsilon_L)(k + d\epsilon_L).$$

In order for δ_L to be well-defined, the discriminant, $h_{d,\kappa}$, must be nonnegative. We can characterize the regions that $h_{d,\kappa}$ is nonnegative by locating its roots which we present in two cases.

Case 1: $h_{d,\kappa}$ has two zeros: $0 < x_1 < x_2$ Consider the case that $h_{d,\kappa}$ has two positive zeros: $0 < x_1 < x_2$. Since $h_{d,\kappa}$ is a 4th-degree polynomial in ϵ_L with nonnegative coefficient, namely d^2 , on the ϵ_L^4 term, and $h_{d,\kappa}(0) = \kappa^2 > 1$, we

can conclude that $h_{d,\kappa}(\epsilon_L)$ is nonnegative on $\epsilon_L \in (0, x_1] \cup [x_2, \infty)$ and negative on $\epsilon_L \in (x_1, x_2)$. See Figure 5.1.

Case 2: $h_{d,\kappa}$ has three or four zeros: $0 < x_1 < x_3 \leq x_4 < x_2$ By the same arguments above, we must have $h_{d,\kappa}(\epsilon_L)$ is nonnegative on $\epsilon_L \in (0, x_1] \cup [x_3, x_4] \cup [x_2, \infty)$ and negative on $\epsilon_L \in (x_1, x_3)$. See Figure 5.1.

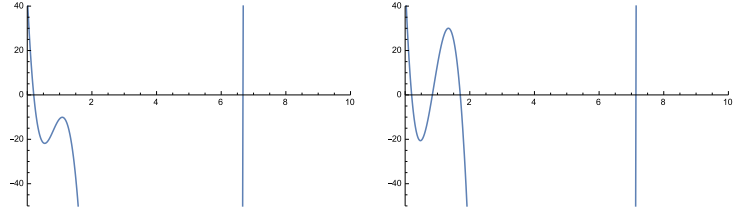


Figure 5.1: Plots of $h_{d,\kappa}$. The left plot illustrates Case 1 above, where $h_{d,\kappa}$ has only two roots (and hence d, κ are not an admissible pair for `RamanujanSparsify`), while the right plot illustrates Case 2 where $h_{d,\kappa}$ has more than two roots (and hence d, κ are admissible).

One can solve for x_1, x_2, x_3, x_4 (when they exist):

$$\begin{aligned} x_1 &= \frac{1}{2d} \left[1 + 2\sqrt{d\kappa} - \kappa + d\kappa - d - \sqrt{(d-1)(\kappa-1)(1+d(\kappa-1) + 4\sqrt{d\kappa} - \kappa)} \right], \\ x_2 &= \frac{1}{2d} \left[1 + 2\sqrt{d\kappa} - \kappa + d\kappa - d + \sqrt{(d-1)(\kappa-1)(1+d(\kappa-1) + 4\sqrt{d\kappa} - \kappa)} \right], \\ x_3 &= \frac{1}{2d} \left[1 - 2\sqrt{d\kappa} - \kappa + d\kappa - d - \sqrt{(d-1)(\kappa-1)(1+d(\kappa-1) - 4\sqrt{d\kappa} - \kappa)} \right], \\ x_4 &= \frac{1}{2d} \left[1 - 2\sqrt{d\kappa} - \kappa + d\kappa - d + \sqrt{(d-1)(\kappa-1)(1+d(\kappa-1) - 4\sqrt{d\kappa} - \kappa)} \right]. \end{aligned}$$

In both Case 1 and Case 2, one can check that substituting $\epsilon_L < x_1$ or $\epsilon_L > x_2$ into (5.6) both yield $\delta_L > (1+\epsilon_L)^{-1}$ which is not permitted to satisfy (5.3). Referring to Figure 5.2, selecting $\epsilon_L < x_1$ results in (ϵ_L, δ_L) contained in the left-most convex

region while selecting $\epsilon_L > x_2$ results in (ϵ_L, δ_L) contained in the right-most convex region, both of which lie above the line $\delta_L = (1 + \epsilon_L)^{-1}$.

However, one can check that if $x_3 \leq x_4$ exist, then substituting $x_3 \leq \epsilon_L \leq x_4$ into (5.6) will indeed yield $\delta_L < (1 + \epsilon_L)^{-1}$. This corresponds to the point (ϵ_L, δ_L) lying in the center convex region illustrated in Figure 5.2. Hence, Case 2 is the *only* possible case to choose parameters that satisfy (5.2) and (5.3) simultaneously.

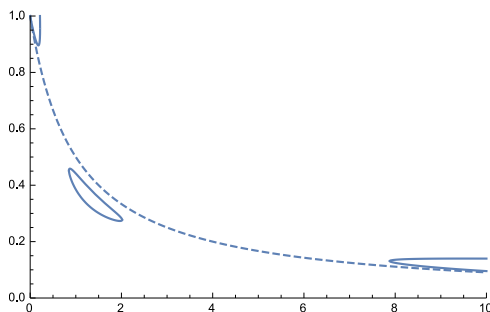


Figure 5.2: The horizontal axis represents the variable ϵ_L and the vertical axis represents δ_L . The dashed line is $\delta_L = (1 + \epsilon_L)^{-1}$. There are three convex regions that determine the pair (ϵ_L, δ_L) by (5.6). The outer two lie above the dashed line and are hence not admissible. Therefore, the choice of (ϵ_L, δ_L) must come from the center region, when it exists.

For a fixed $d > 1$, if $\kappa > 1$ is too small, then $h_{d,\kappa}$ is in Case 1, but one eventually enters Case 2 as κ gets sufficiently large. The breaking point, in other words, the smallest possible κ that enables admissible coefficients, will occur precisely when the local maximum of Figure 5.1 reaches height zero. This happens precisely when

$x_3 = x_4$ which occurs if and only if

$$1 + d(\kappa - 1) - 4\sqrt{d\kappa} - \kappa = 0.$$

This equation has solution

$$\kappa = \kappa_d = \left(\frac{1 + 1/\sqrt{d}}{1 - 1/\sqrt{d}} \right)^2.$$

□

Remark 5.1.4. Notice that the optimal κ is independent of n . The sparsification algorithm depends not on the size of the graph and only the proportion of edges that will remain in H .

Remark 5.1.5. In numerical implementation of `RamanujanSparsify`, the sparsified Laplacian, L_H , often has a tighter spectral approximation than κ_d . But choosing $\kappa < \kappa_d$ will result in `RamanujanSparsify`(L_G, d, κ) to not run.

Remark 5.1.6. Recall that a d -regular graph, R , is a *Ramanujan graph* if

$$\frac{\lambda_{\max}(R)}{\lambda_{\min}(R)} = \kappa_d = \left(\frac{1 + 1/\sqrt{d}}{1 - 1/\sqrt{d}} \right)^2.$$

We remark here that `RamanujanSparsify` does *not* produce a Ramanujan graph; the sparsified graphs produced need not be d -regular. But since they are graphs of dn edges on n vertices, the graph has an average degree of d and achieves the same condition number as the Ramanujan graphs.

It is proved in [55] that any d -regular unweighted graph cannot κ -approximate a complete graph for κ asymptotically better than κ_d . Spielman et. al in [4] conjecture that the same bound holds for *weighted* graphs with *average* degree d . This

conjecture remains open. However, Theorem 5.1.3 proves that RamanujanSparsify cannot produce a weighted d -average degree graph that κ -approximates the complete graph asymptotically better than κ_d . Showing this result for arbitrary graphs of this nature will require a more graph-theoretic argument.

We prove that any κ -approximation of a connected graph will necessarily also be connected.

Corollary 5.1.7. *If G is a connected graph and H is a κ -approximation of G for $0 < \kappa < \infty$, then H is also a connected graph.*

Proof. We know that $\lambda_0^{(G)} = \lambda_0^{(H)} = 0$ and that $\lambda_1^{(G)} > 0$. Suppose that $\lambda_1^{(H)} = 0$. Then $\lambda_1^{(H)}/\lambda_1^{(G)} = 0$, but since H is a κ -approximation of G , then for any k , $\lambda_k^{(H)}/\lambda_k^{(G)} > 1/\kappa$, which contradicts the supposition that $\lambda_1^{(H)} = 0$. Hence H has exactly one zero eigenvalue and therefore has only one connected component. \square

Chapter 6

Graph Conditioning and Optimization

6.1 Condition number and scalable frames

In [4] and works following, Spielman et al. prove how to take a frame $\{f_i\}$ with frame bounds A and B , and rescale it with real nonnegative weights s_i with a large number of the scale equal to zero so that the resulting frame $\{s_i f_i\}$ has frame bounds $A(1 - \epsilon)$ and $B(1 + \epsilon)$.

We now ask a similar yet fundamentally different question. Given a frame $\{f_i\}$ with frame bounds $A < B$, is it possible to rescale the frame (with no restriction on the support of the weights) so that the resulting frame $\{s_i f_i\}$ is a tight frame? Without loss of generality, the tight frame can then be renormalized to form a Parseval frame. Frames that allow such a scaling are called scalable frames

Definition 6.1.1 ([45, Definition 2.1]). A frame $\{f_i\}_{i=1}^m$ in some Hilbert space \mathcal{H} is called a *scalable frame* if there exist nonnegative numbers s_1, \dots, s_m such that $\{s_i f_i\}_{i=1}^m$ is a Parseval frame for \mathcal{H} .

There exists a wealth of literature on classifying scalable frames and measuring how close to tight a frame can be scaled [10, 43, 44]. If a frame is not scalable, then one can measure how “not scalable” the frame is by measuring

$$\min_{s_i \geq 0} \left\| I_N - \sum_{i=1}^m s_i f_i f_i^* \right\|_F,$$

as done in [16], where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.

Recently, the problem of scalable was stated in terms of condition number of the frame.

Definition 6.1.2. The *condition number* of a matrix A , denoted $\kappa(A)$, is defined as the ratio of the largest singular value and the smallest singular value of A , i.e., $\kappa(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$. For a frame in a Hilbert space $\{f_i\}_{i=1}^m \subseteq \mathcal{H}$ with optimal frame bounds A and B , we define the condition number of the frame to be the condition number of its associated frame operator $\kappa(\{f_i\}) := \kappa(S) = B/A$.

In the specific case where X is a symmetric, positive semidefinite, square matrix, we use the definition of condition number from [52] which is consistent with Definition 6.1.2 but is worth explicitly stating here.

Definition 6.1.3. If X is a symmetric positive semidefinite matrix, then the *condition number* is defined as

$$\kappa(X) = \begin{cases} \lambda_{\max}(X)/\lambda_{\min}(X) & \text{if } \lambda_{\min}(X) > 0, \\ \infty & \text{if } \lambda_{\min}(X) = 0 \text{ and } \lambda_{\max}(X) > 0, \\ 0 & \text{if } X \equiv 0. \end{cases}$$

It is proved in [13, Theorem 1.5] that the optimal lower frame bound, A , coincides with the lowest eigenvalue of the frame operator while the optimal upper frame bound, B , equals the largest eigenvalue of the frame operator.

Condition numbers are potentially usefull because a frame is Parseval if and only if its condition number equals 1. Therefore, if a frame $\{f_i\}_{i=1}^m$ is scalable, there

exist scalars $s_i \geq 0$ so that

$$\kappa \left(\sum_{i=1}^m s_i f_i f_i^* \right) = 1.$$

In [12], the authors show that the problem of minimizing the condition number of a frame,

$$\min_{s_i \geq 0} \kappa \left(\sum_{i=1}^m s_i f_i f_i^* \right) = 1,$$

is equivalent to solving the minimization problem

$$\min_{s_i \geq 0} \left\| I_N - \sum_{i=1}^m c_i f_i f_i^* \right\|_2,$$

where $\|\cdot\|_2$ is the operator norm of a matrix.

The task of minimizing the condition number of a matrix is not immediately simple. There is a wealth of literature and algorithms devoted to fast solving of convex programs, see [9] and references therein. In a convex optimization problem, one wants to solve for $x^* = \arg \min_x f(x)$ with for a real convex function $f : X \rightarrow \mathbb{R}$ defined on a convex set X . Convexity of f and X are important because they ensure the following:

1. If x^* is a local minimum of f , then it is a global minimum.
2. The set of all (global) minima is convex.
3. If f is a strictly convex function and a minimum exists, then the minimum is unique.

In addition, the convexity of f and X allows the use of convex analysis to produce fast, efficient algorithmic solvers.

Unfortunately, the condition number function, κ , is not convex. It is however, a quasiconvex function (see [1, Theorem 13.6] for a proof), meaning that its lower level sets form convex sets; that is, the set $\{X : \kappa(X) < a\}$ forms a convex set for any real $a \geq 0$. See [26] and references therein for a survey on some algorithms that can numerically solve certain quasiconvex problems.

While κ is not a convex function, the authors of [48] show that the problem of minimizing condition number is equivalent to solving another problem with convex programming.

Theorem 6.1.4 ([48], Theorem 3.1). *Let $\Omega \subseteq \mathcal{S}^N$ be some nonempty closed convex subset of \mathcal{S}^N , the space of $N \times N$ symmetric matrices and let \mathcal{S}_+^N be the space of symmetric positive semidefinite $N \times N$ matrices. Then the problem of solving*

$$\kappa^* = \inf\{\kappa(X) : X \in \mathcal{S}_+^N \cap \Omega\}$$

is equivalent to the problem of solving

$$\lambda^* = \inf\{\lambda_{\max}(X) : X \in t\Omega, t \geq 0, X \succeq I\}, \quad (6.1)$$

that is, $\lambda^ = \kappa^*$.*

Theorem 6.1.4 has an intuitive interpretation. Suppose $\kappa(X) = \kappa^*$. Then rescaling X by a positive scalar, t , will also scale its eigenvalues by the same factor $1/t$, thus leaving its condition number, $\kappa(X/t)$, unchanged. Therefore, without loss of generality, we can assume that X is rescaled so that $\lambda_{\min}(X/t) \geq 1$ which is imposed in the last condition of (6.1). Once we know that $\lambda_{\min}(X/t)$ is at least 1 then minimizing the condition number of X/t is equivalent to minimizing $\lambda_{\max}(X/t)$ so long as $X/t \in \Omega$ which is guaranteed by the first condition in (6.1).

6.2 Minimizing condition number of graphs

Consider the complete graph K_N . The complete graph is the only graph that has all nonzero eigenvalues equal, i.e., $\lambda_0 = 0$ and $\lambda_1 = \lambda_2 = \cdots = \lambda_{N-1} = N - 1$. It assumes the highest possible for λ_1 , the algebraic connectivity, of a graph on N vertices. The complete graph is the most connected a graph on N vertices can be; the radius of the complete graph is 1 since the distance between any two points, $d(x, y)$, is equal to one.

Recall that for a general graph, the Laplacian can be written as the sum of rank-one matrices $L = \sum_{i=1}^m v_i v_i^*$ where v_i is the i 'th column in the incidence matrix B associated to the i 'th edge in the graph. The problem of Section 6.1 stated in the context of graphs is as follows: How does one find real scalars $s_i \geq 0$ so that the reweighted graph Laplacian $\tilde{L} = \sum_{i=1}^m s_i v_i v_i^*$ minimizes the ratio λ_{N-1}/λ_1 . This is in a sense, reweighting the graph edges so that the resulting graph is most like the complete graph spectrally.

There is the issue of $\lambda_0 = 0$. Any connected graph will have Laplacian eigenvalue $\lambda_0 = 0$ and hence $\kappa(L) = \infty$. We avoid this problem by restricting L to the $(N - 1)$ -dimensional space that is the image of L .

Lemma 6.2.1. *Let G be a connected graph with eigenvalues $\{\lambda_k\}_{k=0}^{N-1}$ and eigenvectors $\{\varphi_k\}_{k=0}^{N-1}$ of the graph Laplacian L . Let $\tilde{\Phi} = [\varphi_1 \varphi_2 \cdots \varphi_{N-1}]$ be the $N \times (N - 1)$ matrix of eigenvectors excluding the constant vector φ_0 . Then the $(N - 1) \times (N - 1)$ matrix*

$$L_0 = \tilde{\Phi}^* L \tilde{\Phi} \tag{6.2}$$

has eigenvalues $\{\lambda_k\}_{k=1}^{N-1}$ and associated orthonormal eigenvectors $\{\tilde{\Phi}^* \varphi_k\}_{k=1}^{N-1}$.

Proof. We first show that $\{\tilde{\Phi}^* \varphi_k\}_{k=1}^{N-1}$ are eigenvectors to L_0 with eigenvalues λ_k .

For any $k = 1, \dots, N - 1$ we have

$$L_0 \tilde{\Phi}^* \varphi_k = \tilde{\Phi}^* L \tilde{\Phi} \tilde{\Phi}^* \varphi_k.$$

Since $\tilde{\Phi}$ is an orthonormal basis for the eigenspace that its vectors span, then $\tilde{\Phi} \tilde{\Phi}^*$ is simply the orthogonal projection onto the eigenspace spanned by $\{\varphi_1, \dots, \varphi_{N-1}\}$.

That is, for any vector f , we have $\tilde{\Phi} \tilde{\Phi}^* f = f - \langle f, \varphi_0 \rangle \varphi_0$, which is simply the function f minus its mean value. For each $k = 1, \dots, N - 1$, the eigenvectors φ_k have zero mean, i.e., $\langle \varphi_k, \varphi_0 \rangle = 0$. Hence $\tilde{\Phi} \tilde{\Phi}^* \varphi_k = \varphi_k$ and therefore

$$L_0 \tilde{\Phi}^* \varphi_k = \tilde{\Phi}^* L \varphi_k = \tilde{\Phi}^* (\lambda_k \varphi_k) = \lambda_k \tilde{\Phi}^* \varphi_k.$$

The orthonormality of the eigenvectors $\{\tilde{\Phi}^* \varphi_k\}_{k=1}^{N-1}$ follows directly from the orthonormality of $\{\varphi_k\}_{k=0}^{N-1}$ and the computation

$$\langle \tilde{\Phi}^* \varphi_k, \tilde{\Phi}^* \varphi_j \rangle = (\tilde{\Phi}^* \varphi_k)^* \tilde{\Phi}^* \varphi_j = \varphi_k^* \tilde{\Phi} \tilde{\Phi}^* \varphi_j = \varphi_k^* \varphi_j = \delta(k, j).$$

□

Unlike the Laplacian, the operator in (6.2) is full rank and its rank equals the rank of the Laplacian. We denote it L_0 because it behaves as the Laplacian after the projection of the function onto the zero'th eigenspace is removed.

We can write the Laplacian in terms of the incidence matrix $L = BB^*$, where recall from Section 1.2.1, $B = [v_1, \dots, v_m]$ where m is the number of edges in the graph and each $v_\ell = e_i - e_j$ for some $(i, j) \in E$. Therefore the operator L_0 can

also be written as one matrix multiplication $L_0 = (\tilde{\Phi}^*B)(\tilde{\Phi}^*B)^*$. As an abuse of language, even though it is not of the particular form defined in Section 1.2.1, we can view the “incidence matrix” of L_0 as $\tilde{\Phi}^*B$ which can also be thought of as the synthesis operator for a full-rank frame in \mathbb{R}^{N-1} . We seek scalars $s_i \geq 1$ so that the rescaled frame $\{\sqrt{s_i}\tilde{\Phi}^*v_i\}_{i=1}^m$ is tight or as close to tight as possible. In terms of matrices, we seek a nonnegative diagonal matrix $X = \text{diag}(\sqrt{s_i})$ so that $\tilde{L}_0 := \tilde{\Phi}^*BX^2B^*\tilde{\Phi}$ has minimal condition number. The resulting graph Laplacian, denoted $\tilde{L}_\kappa = BX^2B^*$, is the operator with minimal condition number, \tilde{L}_0 , without the projection onto $(N - 1)$ eigenspaces, thus acting on the entire N -dimensional space.

We present the pseudocode for the algorithm, `GraphCondition`, that produces \tilde{L}_κ , the Laplacian of the graph that minimizes the condition number of L . In our numerical implementation minimizing condition number, we used the CVX toolbox in MATLAB [21] which is a solver for convex optimization problems.

$L_\kappa = \text{GraphCondition}(L, \Phi, B)$

where L is the Laplacian matrix of the graph G ,

Φ is the $N \times N$ eigenvector matrix of L

B is the incidence matrix of L .

1. Set $\tilde{\Phi} = \Phi(:, 2 : N)$.
2. Use `cvx` to solve for X that minimizes $\lambda_{\max}(\tilde{\Phi}^* B X^2 B^* \tilde{\Phi})$.
subject to: $X \succeq 0$ is diagonal, $\text{trace}(X) \geq t \geq 0$, and $\tilde{\Phi}^* B X^2 B^* \tilde{\Phi} \succeq I$.
3. Renormalize and create $L_\kappa = C \cdot B X^2 B^*$ where C is a scalar chosen so that $\text{trace}(L_\kappa) = \text{trace}(L)$.

Example 6.2.2. We consider the graph G which consists of two complete graphs on 5 vertices that are connected by exactly one edge. The Laplacian for G has eigenvalues $\lambda_1 \approx 0.2984$ and $\lambda_9 \approx 6.7016$, thus giving a condition number of $\kappa(G) \approx 22.45$. We rescale the edges via the `GraphCondition` algorithm and obtained a rescaled weighted graph \tilde{G}_κ which has eigenvalues $\lambda_1 \approx 0.3900$ and $\lambda_{10} \approx 6.991$, thus giving a condition number $\kappa(\tilde{G}_\kappa) \approx 17.9443$.

Both graphs, G and \tilde{G}_κ , are shown in Figure 6.1. The edge bridging the two complete clusters is assigned the highest weight of 1.8473. All other edges emanating from those two vertices are assigned the smallest weights of 0.7389. All other edges not connected to either of the two “bridge” vertices are assigned a weight of 1.1019.

Example 6.2.3. We present condition the graph of the level-5 approximation of the Sierpinski gasket which contains $N = 366$ nodes. The unweighted Laplacian, L ,

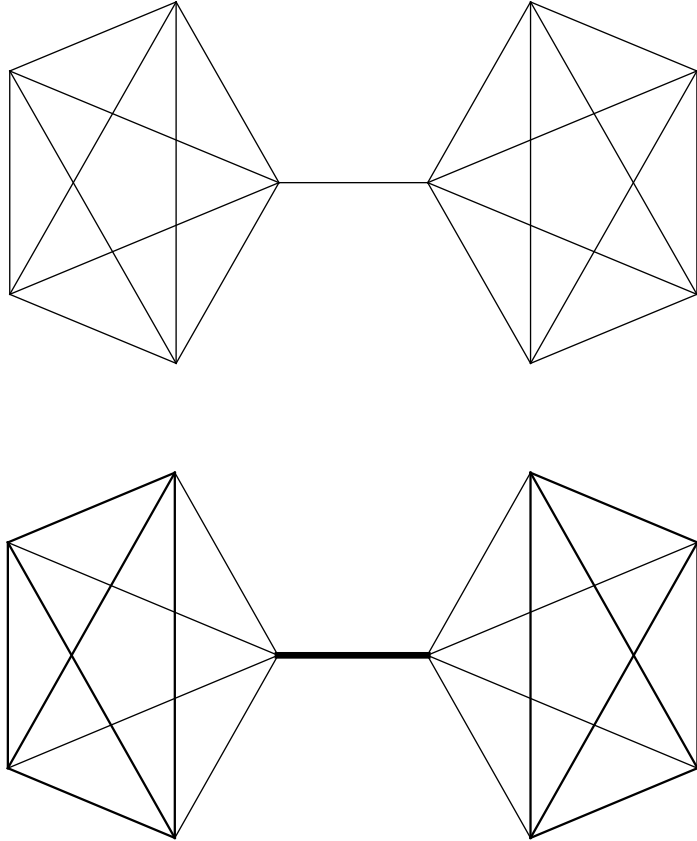


Figure 6.1: Top: The graph G of two complete graphs connected by one edge. Bottom: The conditioned graph with rescaled weights that minimizes the condition number. The width of the edges are drawn to be proportional to the weight assigned to that edge.

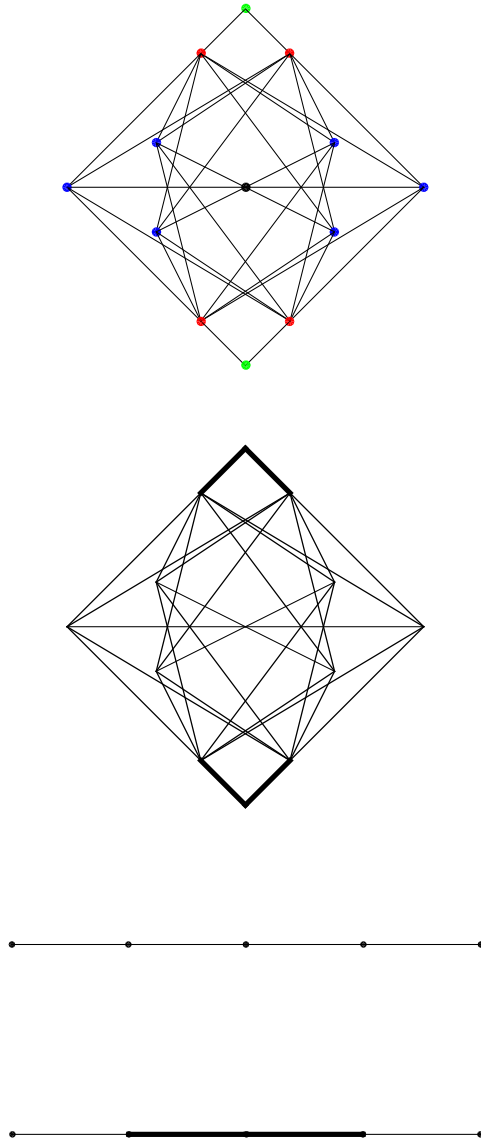


Figure 6.2: Top to bottom: The barren graph, $\text{Barr}(6)$, its conditioned version, The path graph P_5 , and its conditioned version. The width of the edges are drawn to be proportional to the weight assigned to that edge.

has a spectrum with $\lambda_1 \approx 0.0057$ and $\lambda_{N-1} = 6$, thus $\kappa(\text{SG}_5) \approx 1044.9$. Figure 6.3 shows SG_5 and its conditioned version.

We then condition the L using the `GraphCondition` algorithm to obtain a rescaled Laplacian \tilde{L}_κ with eigenvalues $\lambda_1 \approx 0.0065$ and $\lambda_{N-1} \approx 6.7036$ giving $\kappa(\widetilde{\text{SG}}_{5,\kappa}) \approx 1027.6$. Observe that the condition number is only slightly improved by about 1.66%.

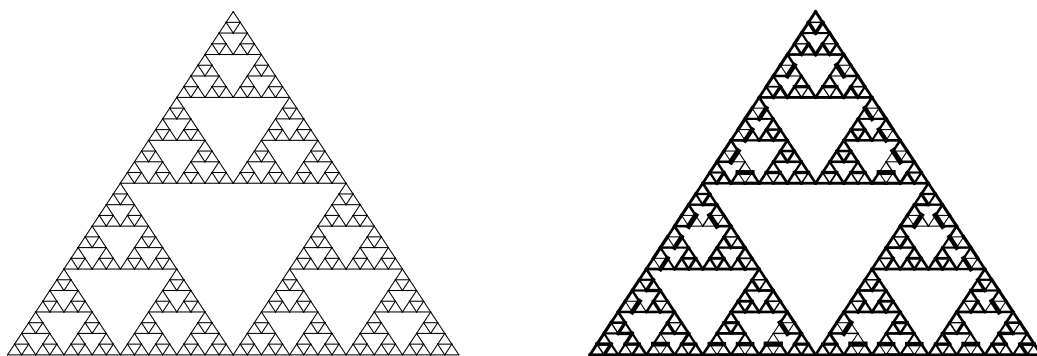


Figure 6.3: Left: The unweighted SG_5 where each edge has a weight of 1 and hence the edges are depicted to be of all the same width. Right: The rescaled and conditioned graph that minimizes condition number. The width of the edges are drawn to be proportional to the weight assigned to that edge.

We show in the following example that the scaling coefficients $\{s_i\}_{i=1}^m$ that minimize the condition number of a graph are not necessarily unique.

Example 6.2.4. Consider the graph G complete graph on four nodes with the edge $(3, 4)$ removed. Then G was rescaled and conditioned via `GraphCondition`;

both graphs are shown in Figure 6.4. The original Laplacian, L , and the rescaled conditioned Laplacian, \tilde{L}_κ , produced by the `GraphCondition` algorithm are given as

$$L = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{bmatrix}, \quad \tilde{L}_\kappa \approx \begin{bmatrix} 2.8406 & -0.6812 & -1.0797 & -1.0797 \\ -0.6812 & 2.8406 & -1.0797 & -1.0797 \\ -1.0797 & -1.0797 & 2.1594 & 0 \\ -1.0797 & -1.0797 & 0 & 2.1594 \end{bmatrix},$$

with spectra

$$\sigma(L) = \{0, 2, 4, 4\}, \quad \sigma(\tilde{L}_\kappa) = \{0, 2.1594, 3.5218, 4.3188\}.$$

Both Laplacians have a condition number $\kappa(L) = \kappa(\tilde{L}_\kappa) = 2$ which shows that the scaling of edges that minimize condition number are not necessarily unique.

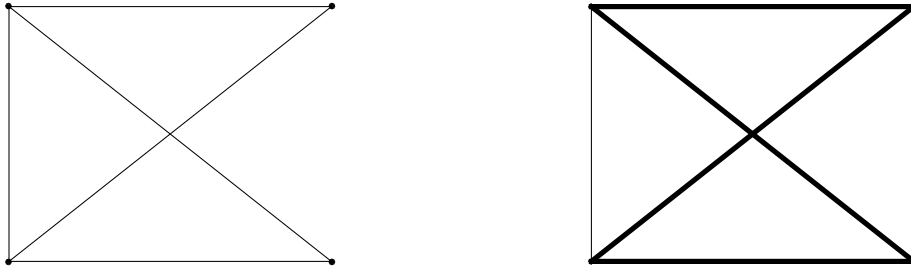


Figure 6.4: The unweighted graph G (left) and its rescaled version \tilde{G}_κ (right) yet both graphs have a condition number equal to 2.

We prove, similar to the result in Corollary 5.1.7, that the `GraphCondition` algorithm will not disconnect a connected graph.

Proposition 6.2.5. *Let $G = G(V, E, \omega)$ be a connected graph and let $\tilde{G}_\kappa = \tilde{G}_\kappa(V, \tilde{E}, \tilde{\omega})$ be the rescaled version of G that minimizes graph condition number. Then \tilde{G}_κ is also a connected graph.*

Proof. Let $\kappa_0 := \kappa(G) \geq 1$ and suppose that \tilde{G}_κ is disconnected. This implies that \tilde{G}_κ has eigenvalue 0 with multiplicity at least 2 (one for each of its connected components). This violates the condition $\tilde{\Phi}^* B X^2 B^* \tilde{\Phi} \succeq I$ in the `GraphCondition` algorithm, which yields the unique minimizer. \square

6.3 Other methods of rescaling graphs

We present in this section other minimization criteria, besides condition number, in which we rescale graphs in order to make the spectrum of the graph closer to the identity.

The first alternative we consider is minimizing the spectral gap of the graph. The spectral gap is defined as $\lambda_{\max} - \lambda_{\min}$, or the length of the smallest interval supporting all nonzero eigenvalues of the Laplacian. The function $\lambda_{\max}(X)$ is convex over the space of symmetric matrices, see [9], and $\lambda_{\min}(X)$ is concave. Hence the spectral gap, $\lambda_{\max}(X) - \lambda_{\min}(X)$, is a convex function and we can numerically minimize it via convex programming. We denote the rescaled graph that minimizes the spectral gap by \tilde{G}_g .

Minimizing the spectral gap of a graph was proposed because it provides equivalent way to characterize scalable frames. A frame is scalable, hence has $\kappa(X) = 1$, if and only if its spectral gap $\lambda_{\max} - \lambda_{\min} = 0$. When the frame is not scalable,

i.e., $\kappa(\tilde{L}_\kappa) > 1$, it is not clear that the scaling that minimizes condition number will necessarily minimize the spectral gap of that frame. In numerical experiments, these minimizers are indeed different and so we consider minimizing condition number as a distinct problem than minimizing condition number as was done in Section 6.2.

The next optimization problems considered are the distance of the rescaled Laplacian to the Laplacian of the complete graph under the Frobenius norm. This problem is again motivated by the idea of rescaling edge weights so that the resulting graph Laplacian is closest to that of the complete graph.

Suppose we were able to rescale the graph G on N vertices to equal a complete graph. Then every vertex on that complete graph must have degree $\bar{d} = \text{trace}(L_G/N)$. Recall in Section 6.2 that for a connected graph's Laplacian, L , we can consider the operator, L_0 acting as the Laplacian restricted to the $(N - 1)$ -dimensional space excluding the zero eigenspace. For the complete graph, K_N , it's restricted Laplacian is equivalent to $\bar{d}I_{N-1}$. Therefore, in numerical implementation, the objective function that we minimize is $\|L_0 - \bar{d}I_{N-1}\|$ under the Frobenius norm. Minimization of a norm is a convex problem and can also be solved using existing convex programs. We let \tilde{G}_F denote the rescaled graph that minimizes the objective function with the Frobenius norm.

Example 6.3.1. We present numerical results of each of the graph rescaling techniques for the double cluster graph shown in Figure 6.1. Each of the rescaled graphs are pictured in Figure 6.5 and numerical data is summarized in Table 6.1.

We remark that the graph condition number of \tilde{G}_κ is reduced by only about

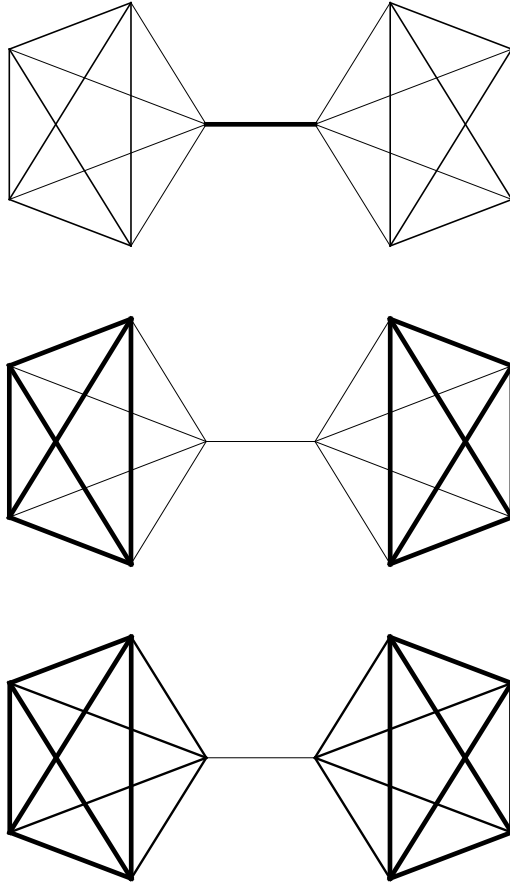


Figure 6.5: From top to bottom: \tilde{G}_κ , \tilde{G}_g , \tilde{G}_F , which minimize the condition number, spectral gap, and the Frobenius distance to identity, respectively.

	G	\tilde{G}_κ	\tilde{G}_g	\tilde{G}_F
κ	22.4555	17.9443	22.8786	24.6280
$\lambda_{N-1} - \lambda_1$	6.4031	6.6091	5.1509	5.6526
$\ L_0 - \bar{d}I_{N-1}\ _F$	12.2246	12.3098	13.0895	12.1791

Table 6.1: Comparison of condition number, spectral gap, and Frobenius distance to identity of the cluster graph, G , shown in Figure 6.1 and its rescaled versions, respectively.

20%. A source of future work would be to quantify lower bounds on rescaled graphs.

Example 6.3.2. We repeat the simulation shown in Example 6.2.2 for the level-5 graph approximation to the Sierpinski gasket, SG_5 and present the results in Figure 6.6 and Table 6.2. We observe that the condition number minimizer reduces the condition number by less than 2%.

Figures 6.7 and 6.8 illustrate the comparison of these rescaling methods for the path graph P_5 and the barren graph, $Barr(6)$, that was introduced in Section 3.1.1. The interpretation of the resulting numerical implementation of these algorithms still requires development. The data suggests that graph conditioning (minimizing the condition number) produces the best approximation to the complete graph, and we will discuss this interpretation further in Section 6.4. Meanwhile, minimizing the spectral gap and Frobenius distance to K_N does not appear to provide good approximations to the complete graph. One can observe in Figures 6.5, 6.6, and

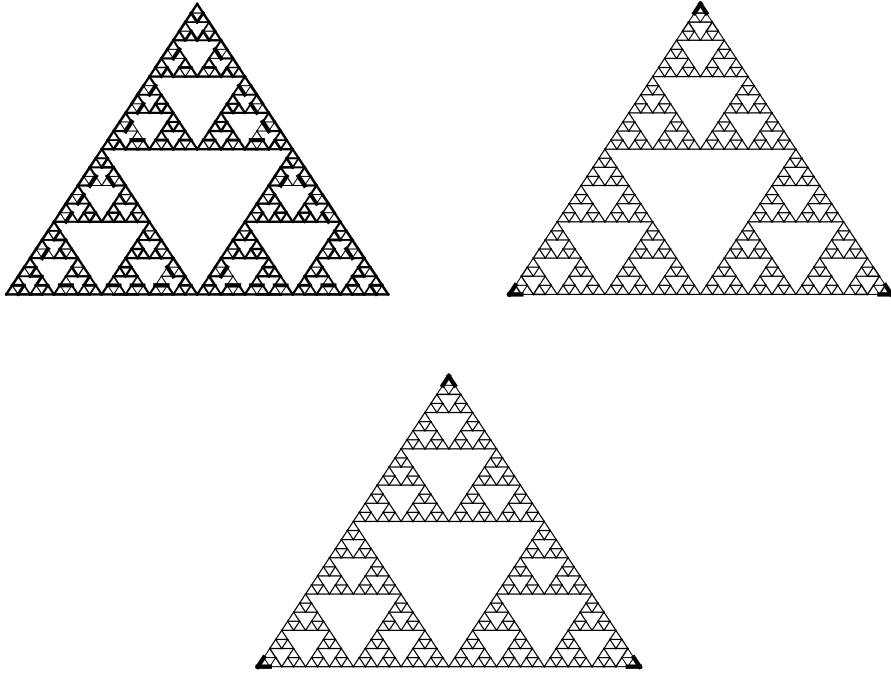


Figure 6.6: Left to right: \tilde{G}_κ , \tilde{G}_g , \tilde{G}_F , which minimize the condition number, spectral gap, and Frobenius distance to identity, respectively.

	G	\tilde{G}_κ	\tilde{G}_g	\tilde{G}_F
κ	1044.9	1027.6	1045.0	1048.4
$\lambda_{N-1} - \lambda_1$	5.9943	6.6971	5.9737	6.0000
$\ L_0 - \bar{d}I_{N-1}\ _F$	85.3112	87.9702	85.7364	85.2882

Table 6.2: Comparison of condition number, spectral gap, and Frobenius distance to identity of the level-5 graph approximation to the Sierpinski gasket, SG_5 , and its rescaled versions, respectively.

6.7 that these two methods assign high weights to the edges that are furthest away from the rest of the graph. And yet, as previously discussed, there are situations in which both optimization methods give the same optimal edge weights. Then, there are graphs with high symmetries, such as the barren graph in Figure 6.8 whose rescaled versions are nearly indistinguishable yet their Laplacians do have nuances and fundamental differences. Interpreting these results and developing theory to explain the numerics is a source of future work that we hope to settle.

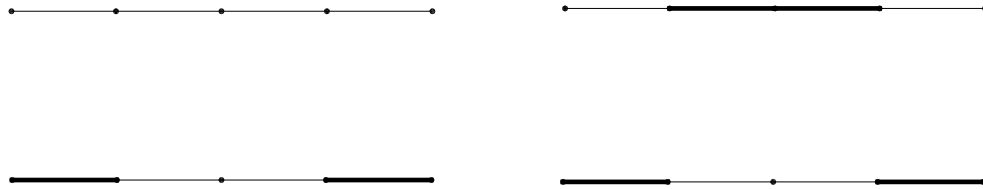


Figure 6.7: Left to right: The path graph on 5 vertices, $G = P_5$, and its rescaled versions \tilde{G}_κ , \tilde{G}_g , and \tilde{G}_f .

6.4 Interpretation of graph conditioning

As discussed in the motivation of this section, reducing the condition number of a graph makes the graph more “complete”, that is, more like the complete graph in terms of its spectrum. Since the algebraic connectivity λ_1 is as great as possible, it is the only graph for which $\lambda_1 = \lambda_{N-1}$, the graph is the most connected a graph can possibly be, and as such the distance between any two points is minimal. As

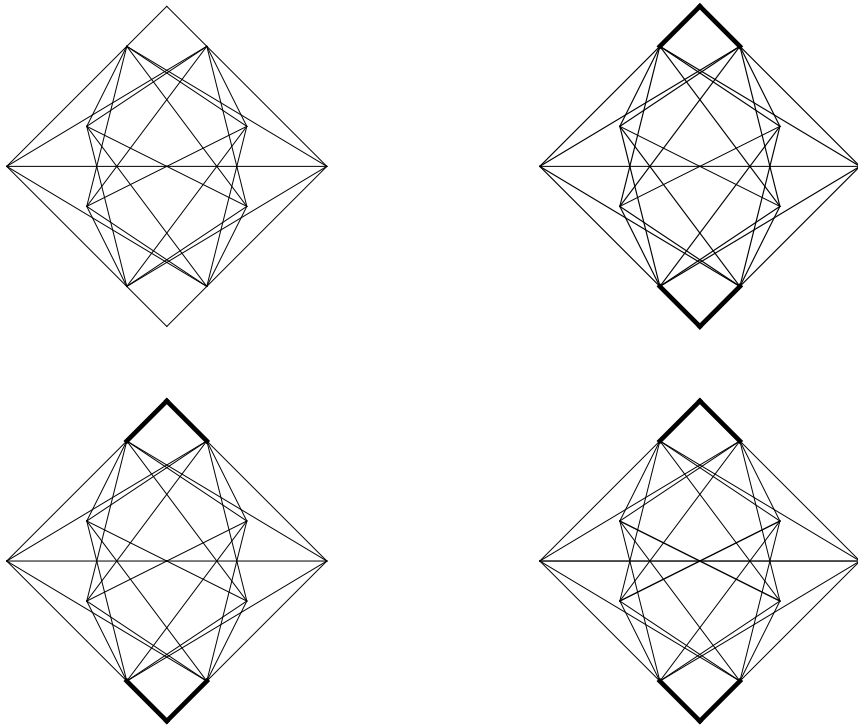


Figure 6.8: Left to right: The barren graph of 13 vertices, $G = \text{Barr}(6)$, and its rescaled versions \tilde{G}_κ , \tilde{G}_g , and \tilde{G}_f .

previously discussed, the effective resistance is a natural metric on graphs and one can compute that for any two distinct vertices, i and j , on the complete graph on N vertices we have

$$\begin{aligned}
R(i, j) &= \sum_{k=1}^{N-1} \frac{1}{\lambda_k} (\varphi_k(i) - \varphi_k(j))^2 = \frac{1}{N} \sum_{k=1}^{N-1} (\varphi_k(i) - \varphi_k(j))^2 \\
&= \frac{1}{N} (e_i - e_j)^* \Phi \Phi^* (e_i - e_j) = \frac{1}{N} (e_i - e_j)^* (e_i - e_j) \\
&= \frac{1}{N} \|e_i - e_j\|^2 = \frac{2}{N}.
\end{aligned}$$

Conjecture 6.4.1. *The process of conditioning a graph reduces the average resistance between any two vertices on the graph.*

The intuition behind Conjecture 6.4.1 can be motivated by studying the quantity $\sum_{k=1}^{N-1} 1/\lambda_k$. Consider a sequence of positive numbers $\{a_k\}_{k=1}^N$ with average $\bar{a} = 1/N \sum_{k=1}^N a_k$. Then since the function $h(t) = 1/t$ is continuous and convex on the set of positive numbers, it is also midpoint convex on that set, i.e.,

$$\frac{N}{\bar{a}} = Nh(\bar{a}) \leq \sum_{k=1}^N h(a_k) = \sum_{k=1}^N \frac{1}{a_k}.$$

With this fact, let $\{\lambda_k\}_{k=1}^{N-1}$ denote the eigenvalues of connected graph G and $\{\tilde{\lambda}_k\}_{k=0}^{N-1}$ denote the eigenvalues of the conditioned graph \tilde{G}_κ , both satisfying $\bar{\lambda} = 1/N \sum_{k=1}^{N-1} \lambda_k = 1/N \sum_{k=1}^{N-1} \tilde{\lambda}_k$. Since \tilde{G}_κ is better conditioned than G , then $\left\| \sum_{k=1}^{N-1} \tilde{\lambda}_k - \bar{\lambda} \right\| \leq \left\| \sum_{k=1}^{N-1} \lambda_k - \bar{\lambda} \right\|$. In other words, the eigenvalues $\{\tilde{\lambda}_k\}_{k=1}^{N-1}$ are closer to the average $\bar{\lambda}$ than the eigenvalues $\{\lambda_k\}_{k=1}^{N-1}$ are. Hence

$$\sum_{k=1}^{N-1} \frac{1}{\tilde{\lambda}_k} \leq \sum_{k=1}^{N-1} \frac{1}{\lambda_k}. \tag{6.3}$$

Equation (6.3) resembles the effective resistance $R(i, j) = \sum_{k=1}^{N-1} 1/\lambda_k (\varphi_k(i) - \varphi_k(j))^2$ except for the term $(\varphi_k(i) - \varphi_k(j))^2$. This term will be difficult to account for since

little is known about the eigenvectors of \tilde{G}_κ . Analysis of eigenvectors of perturbed matrices is a widely open area of research and results are very limited, see [18, 40, 65, 68].

We remark that Conjecture 6.4.1 claims that conditioning a graph will reduce the average effective resistance between points; it is not true that the resistance between all points will be reduced. If the weight on edge (i, j) is reduced, then its effective resistance between points i and j is increased. Since we impose that the trace of the Laplacians be preserved, if any edge weights are increased, then by conservation at least one other edge's weight must be decreased. The vertex pairs for those edges will then have an increased effective resistance between them.

While we lack the theoretical justification, numerical simulations support Conjecture 6.4.1. Figure 6.9 shows the effective resistance matrix for SG_5 and its conditioned counterpart, both shown in Figure 6.3. To emphasize their relation, Figure 6.9 also shows the difference $R - \tilde{R}_\kappa$ which has mostly positive values. The average value of $R - \tilde{R}_\kappa$ is approximately 1.0575.

The authors of [34] approach a similar way. They propose using convex optimization to minimize the total effective resistance of the graph,

$$R_{tot} = \sum_{i,j=1}^N R(i, j).$$

They show that the optimization problem is related to the problem of reweighting edges to maximize the algebraic connectivity λ_1 .

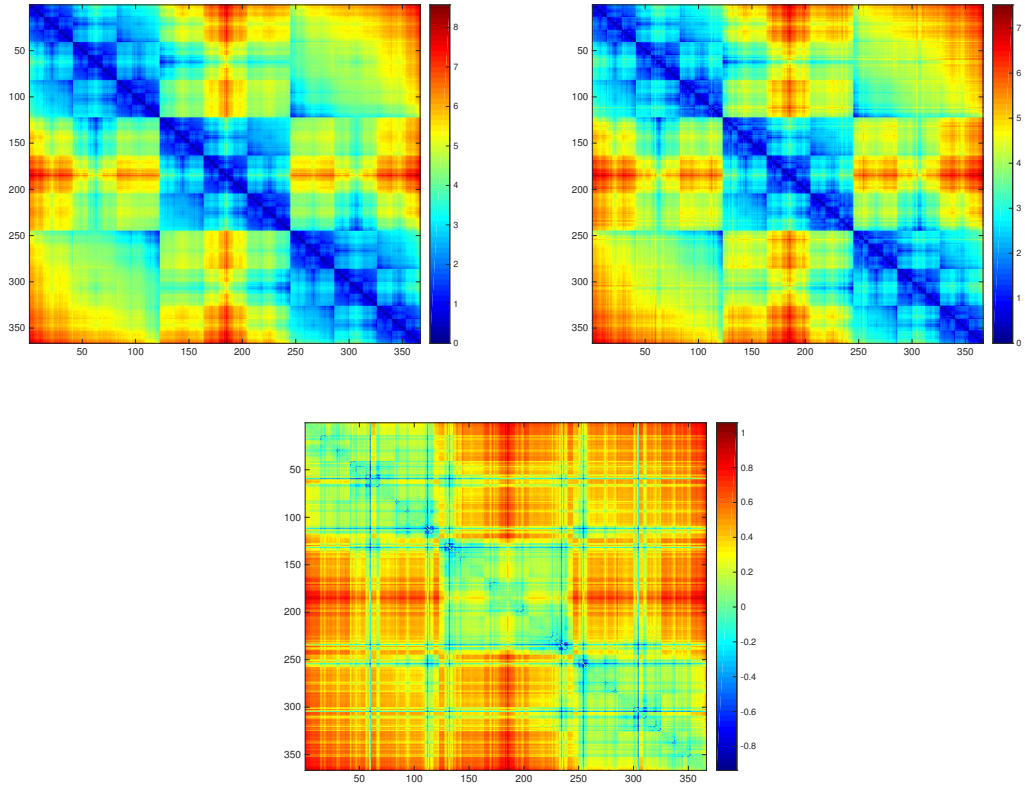


Figure 6.9: The effective resistance matrix, R , for SG_5 (left) and the effective resistance matrix, \tilde{R}_κ , for its conditioned counterpart (bottom). The right image shows the difference $R - \tilde{R}_\kappa$ for which most values are positive.

Bibliography

- [1] O. Axelsson. *Iterative Solution Methods*. Cambridge University Press, 1996.
- [2] R. Bapat and S. Pati. Algebraic connectivity and the characteristic set of a graph. *Linear and Multilinear Algebra*, 45(2-3):247–273, 1998.
- [3] S. Barik, S. Fallat, and S. Kirkland. On Hadamard diagonalizable graphs. *Linear Algebra and its Applications*, 435(8):1885–1902, 2011.
- [4] J. Batson, D. A. Spielman, and N. Srivastava. Twice-Ramanujan sparsifiers. *SIAM Review*, 56(2):315–334, 2014.
- [5] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
- [6] J. J. Benedetto. *Harmonic Analysis and Applications*, volume 23. CRC Press, 1996.
- [7] J. J. Benedetto and W. Czaja. *Integration and Modern Analysis*. Springer Science & Business Media, 2010.
- [8] J. J. Benedetto and S. Li. The theory of multiresolution analysis frames and applications to filter banks. *Applied and Computational Harmonic Analysis*, 5(4):389–427, 1998.
- [9] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [10] J. Cahill and X. Chen. A note on scalable frames. In *10th International Conference on Sampling Theory and Applications (SampTA 2013)*, pages 93–96, Bremen, Germany, July 2013.
- [11] P. G. Casazza. The Kadison–Singer and Paulsen problems in finite frame theory. In *Finite Frames*, pages 381–413. Springer, 2013.
- [12] P. G. Casazza and X. Chen. Frame scalings: A condition number approach. *arXiv preprint arXiv:1510.01653*, 2015.
- [13] P. G. Casazza and G. Kutyniok. Introduction to finite frames. In P. G. Casazza and G. Kutyniok, editors, *Finite Frames, Theory and Applications*, pages 1–53. Springer-Birkhäuser, New York, 2013.
- [14] P. G. Casazza and J. C. Tremain. The Kadison–Singer problem in mathematics and engineering. *Proceedings of the National Academy of Sciences of the United States of America*, 103(7):2032–2039, 2006.

- [15] A. K. Chandra, P. Raghavan, W. L. Ruzzo, R. Smolensky, and P. Tiwari. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity*, 6(4):312–340, 1996.
- [16] X. Chen, G. Kutyniok, K. A. Okoudjou, F. Philipp, and R. Wang. Measures of scalability. *Information Theory, IEEE Transactions on*, 61(8):4410–4423, 2015.
- [17] F. R. Chung. *Spectral Graph Theory*, volume 92. American Mathematical Soc., 1997.
- [18] A. Cloninger. Exploiting Data-Dependent Structure for Improving Sensor Acquisition and Integration. *Ph.D Thesis, University of Maryland, College Park*, 2014.
- [19] R. R. Coifman and M. J. Hirn. Diffusion maps for changing data. *Applied and Computational Harmonic Analysis*, 36(1):79–107, 2014.
- [20] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.
- [21] I. CVX Research. CVX: Matlab software for disciplined convex programming, version 2.0. <http://cvxr.com/cvx>, Aug. 2012.
- [22] W. Czaja and M. Ehler. Schroedinger eigenmaps for the analysis of bio-medical data. *IEEE Trans Pattern Anal Mach Intell.*, 35(5):1274–80, 2013.
- [23] T. Davis. UF Sparse Matrix Collection. <http://www.cise.ufl.edu/research/sparse/matrices/Gleich/>.
- [24] P. G. Doyle and J. L. Snell. *Random walks and electric networks*. Number 22 in Carus Mathematical Monographs. Mathematical Association of America, 1984.
- [25] W. Ellens, F. Spieksma, P. Van Mieghem, A. Jamakovic, and R. Kooij. Effective graph resistance. *Linear Algebra and its Applications*, 435(10):2491–2506, 2011.
- [26] D. Eppstein. Quasiconvex programming. *Combinatorial and Computational Geometry*, 52:287–331, 2005.
- [27] S. M. Fallat, S. J. Kirkland, J. J. Moliterno, and M. Neumann. On graphs whose laplacian matrices have distinct integer eigenvalues. *Journal of Graph Theory*, 50(2):162–174, 2005.
- [28] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.
- [29] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973.

- [30] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975.
- [31] J. Friedman and J.-P. Tillich. Calculus on graphs. *arXiv preprint cs/0408028*, 2004.
- [32] J. Friedman and J.-P. Tillich. Wave equations for graphs and the edge-based laplacian. *Pacific Journal of Mathematics*, 216(2):229–266, 2004.
- [33] M. Gavish, B. Nadler, and R. R. Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 367–374, 2010.
- [34] A. Ghosh, S. Boyd, and A. Saberi. Minimizing effective resistance of a graph. *SIAM review*, 50(1):37–66, 2008.
- [35] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.
- [36] A. Hedayat, W. Wallis, et al. Hadamard matrices and their applications. *The Annals of Statistics*, 6(6):1184–1238, 1978.
- [37] S. M. Heilman and R. S. Strichartz. Localized eigenfunctions: Here you see them, there you dont. *Notices of the AMS*, 57(5):624–629, 2010.
- [38] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 2012.
- [39] W. Imrich, S. Klavzar, and D. F. Rall. *Topics in graph theory: Graphs and their Cartesian product*. CRC Press, 2008.
- [40] T. Kato. *Perturbation Theory for Linear Operators*, volume 132. Springer Science & Business Media, 1976.
- [41] J. Kigami. *Analysis on Fractals*, volume 143. Cambridge University Press, 2001.
- [42] C. Kuratowski. Sur le problème des courbes gauches en topologie. *Fundamenta mathematicae*, 1(15):271–283, 1930.
- [43] G. Kutyniok, K. A. Okoudjou, and F. Philipp. Preconditioning of frames. In *SPIE Optical Engineering+ Applications*, pages 88580G–88580G. International Society for Optics and Photonics, 2013.
- [44] G. Kutyniok, K. A. Okoudjou, and F. Philipp. Scalable frames and convex geometry. *Contemp. Math*, 626:19–32, 2014.

- [45] G. Kutyniok, K. A. Okoudjou, F. Philipp, and E. K. Tuley. Scalable frames. *Linear Algebra and its Applications*, 438(5):2225–2238, 2013.
- [46] Y. T. Lee, R. Peng, and D. A. Spielman. Sparsified cholesky solvers for sdd linear systems. *arXiv preprint arXiv:1506.08204*, 2015.
- [47] Y. T. Lee and H. Sun. Constructing linear-sized spectral sparsification in almost-linear time. *arXiv preprint arXiv:1506.03261*, 2015.
- [48] Z. Lu and T. K. Pong. Minimizing condition number via convex programming. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1193–1211, 2011.
- [49] A. Marcus, D. A. Spielman, and N. Srivastava. Interlacing families I: Bipartite Ramanujan graphs of all degrees. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 529–537. IEEE, 2013.
- [50] A. Marcus, D. A. Spielman, and N. Srivastava. Interlacing families II: Mixed characteristic polynomials and the Kadison-Singer problem. *Annals of Mathematics*, 182(1):327–350, 2015.
- [51] A. W. Marcus, D. A. Spielman, and N. Srivastava. Ramanujan graphs and the solution of the kadison-singer problem. In *2014 International Congress of Mathematicians*, 2014.
- [52] P. Maréchal and J. J. Ye. Optimizing condition numbers. *SIAM Journal on Optimization*, 20(2):935–947, 2009.
- [53] J. Needleman, R. S. Strichartz, A. Teplyaev, and P.-L. Yung. Calculus on the Sierpinski gasket I: polynomials, exponentials and power series. *Journal of Functional Analysis*, 215(2):290–340, 2004.
- [54] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.
- [55] A. Nilli. On the second eigenvalue of a graph. *Discrete Mathematics*, 91(2):207–210, 1991.
- [56] K. A. Okoudjou and R. S. Strichartz. Asymptotics of eigenvalue clusters for Schrödinger operators on the Sierpinski gasket. *Proceedings of the American Mathematical Society*, 135(8):2453–2459, 2007.
- [57] K. A. Okoudjou, R. S. Strichartz, and E. K. Tuley. Orthogonal polynomials on the Sierpinski gasket. *Constructive Approximation*, 37(3):311–340, 2013.
- [58] N. Sharon and Y. Shkolnisky. A class of Laplacian multiwavelets bases for high-dimensional data. *Applied and Computational Harmonic Analysis*, 38(3):420–451, 2015.
- [59] D. I. Shuman, B. Ricaud, and P. Vandergheynst. Vertex-frequency analysis on graphs. *Applied and Computational Harmonic Analysis*, 40(2):260–291, 2016.

- [60] D. A. Spielman. Spectral graph theory. *Lecture Notes, Yale University*, pages 740–776, 2009.
- [61] D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.
- [62] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 81–90. ACM, 2004.
- [63] D. A. Spielman and S.-H. Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.
- [64] D. A. Spielman and S.-H. Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26, 2013.
- [65] G. W. Stewart. *Matrix perturbation theory*. Academic Press, Inc., 1990.
- [66] R. S. Strichartz. How to make wavelets. *American Mathematical Monthly*, pages 539–556, 1993.
- [67] R. S. Strichartz. *Differential Equations on Fractals: a Tutorial*. Princeton University Press, 2006.
- [68] T. Tao. *Topics in Random Matrix Theory*, volume 132. American Mathematical Soc., 2012.
- [69] J. C. Urschel and L. T. Zikatanov. Spectral bisection of graphs and connectedness. *Linear Algebra and its Applications*, 449:1–16, 2014.
- [70] N. K. Vishnoi. Laplacian solvers and their algorithmic applications. *Theoretical Computer Science*, 8(1-2):1–141, 2012.
- [71] K. Wagner. Über eine eigenschaft der ebenen komplexe. *Mathematische Annalen*, 114(1):570–590, 1937.