Discrete Integral Operators on Graphs and Multiscale Transforms on Simplicial Complexes

By

EUGENE SHV RTS
DISSERT TION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

pplied Mathematics

in the

OFFICE OF GR DU TE STUDIES

of the

UNIVERSITY OF C LIFORNI

D VIS

pproved:

Naoki Saito, Chair

James Bremer

Jesus De Loera

Committee in Charge

2023

Contents

Discrete Integral Operators on Graphs and Multiscale Transforms on Simplicial Complexes

bstract

s network analysis tools have become more powerful over the last decade, the need has arisen for new multiscale signal processing tools, both for robust understanding of the intrinsic geometry underlying data, and for higher-order signals, incorporating not only data which lives on the vertices of a graph, but perhaps on its edges, or the faces of a triangular mesh of a manifold. In this dissertation, first we develop a discrete integral operator suitable for spectral embedding and partitioning of graphs, by carefully studying the development of graph Laplacian techniques from their continuous analogues on domains, and applying those developments to integral operators which commute with the continuous Laplacian. We demonstrate the operator's effectiveness at distinguishing underlying geometry in scattered data, and efficient sparse techniques for performing partitioning using it. Next, we present extensions of two powerful multiscale graph signal transforms for analyzing signals defined on the κ-dimensional simplices of a simplicial complex. The previous Hierarchical Graph Laplacian Eigen Transform (HGLET) generalizes the block DCT to the graph setting, and our Hierarchical κ-Laplacian Eigen Transform (κ-HGLET) generalizes further to the simplicial complex setting. Likewise, for the previous Generalized Haar-Walsh Transform (GHWT) which generalizes the Haar-Walsh wavelet packet transform, we propose the κ-Generalized Haar-Walsh Transform (κ-GHWT). The key idea is to use the Hodge Laplacians and their variants for hierarchical bipartitioning of the κ-dimensional simplices in a given simplicial complex, and then building localized basis functions on these partitioned subsets. We demonstrate the usefulness of the κ-HGLET and κ-GHWT on both illustrative synthetic examples and real-world simplicial complexes generated from a co-authorship/citation dataset.

cknowledgments

I'm extraordinarily grateful to my advisor, Professor Naoki Saito. He has been a stalwart, fair, patient, and highly-skilled mentor, an excellent role model, and a source of inspiration. He has stayed highly involved with my research across two separate stints of my PhD work, constantly opened new directions and doors for me, and gave me consistent opportunities to advance my research and professional career. I feel incredibly fortunate for his attention, support, and positive influence over me.

I'd like to thank Professor Stefan Schonsheck for his advice, guidance, diligent experimental and exploratory work, and having provided a strong hand in shaping my current research direction. Working together over the last year and a half has been refreshing and rewarding.

I'd like to thank my parents, who housed me when I returned to finish my degree, and have been endless sources of encouragement, both material and emotional. There's nothing like home cooking for morale. I feel grateful for having had this opportunity to be closer to family as well.

This research was partially supported by Professor Saito's NSF grants DMS-1418779, 1934568, and 1912747.

Together with Professor Saito, I'd like to thank my committee members Professors James Bremer and Jesus De Loera for their helpful and extremely prompt feedback.

I'd like to thank my colleagues in Prof. Saito's lab, Jeff Irion, Chelsea Weaver, lex Berrian, David Weber, Haotian Li, and Yiqun Shao for participating in discussion around my research, and offering helpful advice. The culture of software development in our lab is also top-notch, and encouraged me both to grow my skills, and prepare software I can release with my dissertation.

Thank you Dmitry Shemetov, for skulls, drums, and other absolutely vital grad school resources.

Thanks to my classmates Eric Samperton and Yoni ckerman for encouraging me through the tumultuous first few years we shared together.

Thanks to everyone who's believed in me, and pushed me forward.

CH PTER 1

Introduction

We live in a fast-paced, tightly networked world, which continues to demand faster, greater-scale, more multi-faceted analysis of the data that drive it. For conventional digital signals and images sampled on regular lattices, *multiscale basis dictionaries*, i.e., *wavelet packet dictionaries* including *wavelet bases*, *local cosine dictionaries*, and their variants (see, e.g., [**99**, Chap. 4, 7], [**52**, Chap. 6, 7], [**67**, Chap. 8]), have a proven track record of success: *JPEG 2000* Image Compression Standard [**81**, Sec. 15.9]; *Modified Discrete Cosine Transform* (MDCT) in MP3 [**81**, Sec. 16.3]; *discriminant feature extraction* for signal classification [**76**, **77**, **78**], just to name a few. Considering the abundance of data measured on graphs and networks and the increasing importance to analyze such data (see, e.g., [**15**, **29**, **66**, **68**, **88**]), it is quite natural to lift/generalize these dictionaries to the graph setting. From the perspective of harmonic analysis, those tools have evolved to become more flexible and precise in response, both in development of localized overcomplete dictionaries, and in techniques for searching, optimizing, and applying them. Our contribution builds on a series of developments in Prof. Saito's lab started in [**47**, **48**, **49**, **51**], and continued in many other directions, which use hierarchical bipartition trees to structure graphs into the analogue of spatial regions at multiple scales, upon which well-localized orthogonal bases and overcomplete dictionaries can be constructed, and used to analyze signals on the *vertices* of the graph. We focus on two distinct areas: primarily, to extend these constructions to *oriented simplicial complexes*, especially with the difficulties introduced by orientation, and multi-way relationships, and secondarily, to further develop the theory of discrete integral operators towards bipartitioning methods more sensitive to latent geometry, and more suitable for directed graphs.

Graph-based methods for analyzing data have been widely adopted in many domains; see, e.g., [**11**, **28**, **69**]. Often, these graphs are fully defined by data (such as a graph of social media "friends"), but they can also be induced through the persistence homology of generic point clouds [**13**]. In either case, the vast majority of these analytical techniques deal with signals which are defined on

the vertices of a given graph. More recently, there has been a surge in interest in studying signals defined on edges, triangles, and higher-dimensional substructures within the graph [**7**, **13**, **14**, **35**]. fundamental tool employed for analyzing these signals, the *Hodge Laplacian*, has been studied in the context of differential geometry for over half a century but has only recently entered the toolbox of applied mathematics. Spectral analysis of signals on simplicial complexes using the Hodge Laplacian is a rapidly developing field, and recent work includes fundamental research into the behavior of random walks analogous to the establishment of PageRank [**82**]. This rise in popularity is largely due to the adaptation of discrete differential geometry [**23**] in applications in computer vision [**65**, **73**], statistics [**53**], topological data analysis [**14**, **84**], and network analysis [**82**, **83**].

One of the key challenges to applying wavelets and similar constructions to vertex-based graph signals is that graphs lack a natural translation operator, which prevents the construction of convolutional operators and traditional Littlewood-Paley theory [**49**, **60**, **85**]. This challenge is also present, and magnified, for general κ-dimensional simplices. One method for overcoming this difficulty is to perform convolution solely in the "frequency" domain and define wavelet-like bases entirely in the coefficient space of the Laplacian (or in this case Hodge Laplacian) transform. Following this line of research, there have been several approaches to defining wavelets [**72**] and convolutional neural networks [**30**] in which the input signal is transformed in a series of coefficients in the eigenspace of the Hodge Laplacian. Unfortunately, the atoms (or basis vectors) generated by these methods are not always locally supported, and it can be difficult to interpret their role in analyzing a given graph signal.

n alternative path to the creation of wavelet-like dictionaries and transforms is to first develop a hierarchical block decomposition of the domain and then use this to develop multiscale transforms [**47**, **48**, **80**]. These techniques rely on recursively computing bipartitions of the domain and then generating localized bases on the subsets of the domain. In this dissertation, we propose a simplicial analog to the Fielder vector [**31**, **41**] to solve a relaxed version of a variety of cut-problems for κ-simplices directly analogous to Ratio Cut [**38**] and Normalized Cut [**87**], which we can apply iteratively to develop a hierarchical bipartition of the κ-dimensional simplices in a simplicial complex. From here, we are able to apply the general scheme of [**48**] and [**47**] to develop the *Hierarchical κ-Laplacian Eigen Transform* and the *κ-Generalized Haar-Walsh Transform*, respectively,

for a given collection of simplices of an arbitrarily high order. s a result, we can also generate orthonormal Haar bases, orthonormal Walsh bases, as well as data-adaptive orthonormal bases using the best-basis selection method [**20**].

1.1. Organization and Contributions

This dissertation is organized as follows. In Chapter 2, we review the foundational concepts required for multiscale transforms, including wavelet packets, recursive graph partitioning, and the fundamentals of simplicial complexes. In Chapter 3, we describe the setting of commuting integral and differential operators which provides the continuous backdrop over much of the discrete spectral theory used throughout, and construct a new bipartitioning method for graphs through a new interpretation of a discrete integral operator. Chapters 4-6 are based on the preprint [**79**] co-authored with Prof. Naoki Saito and Prof. Stefan Schonsheck, and much of that material is developed similarly here. In Chapter 4, we define the Fiedler vector for several variations of the Hodge Laplacian, and extend the recursive bipartitioning scheme to the κ-simplicial setting. In Chapter 5, we define and describe the properties of the κ-GHWT and κ-HGLET multiscale transforms. In Chapter 6, we perform several numerical experiments in approximation and classification, demonstrating the performance of our multiscale transforms on real and synthetic datasets, and compare them to existing ones. Finally, we conclude with Chapter 7 discussing our potential future work.

Several of the new contributions in this dissertation were developed with the authors of [**79**] in the preparation of our preprint, including the construction of the κ-Haar basis, and κ-GHWT and κ-HGLET dictionaries, employing these dictionaries in best-basis search, and a number of experiments. My own contributions include:

- novel perspective on simplex orientation vis a vis natural orientation, which leads directly to the construction of the Fiedler vector for κ-Laplacians in Chapter 4;
- n approximation scheme for hierarchical partitioning called submatrix partitioning, introduced in Section 2.3, which yields significant efficiency and stability in the simplicial complex setting in Section 5.3;
- rigorous justification for the usage of the graph distance matrix in constructing discrete integral operators for spectral embedding and partitioning of graphs in Chapter 3;

- demonstration of sparse algorithms, regularization properties, and spectral geometry applications which make the discrete integral operator comparable to the usual Laplacian, in Section 3.3.

1.2. Notation

Notation	Description
κ	the order or dimensionality of a simplex
C_κ	a κ-region, or the set of all κ-simplices in a complex
j	the scale/resolution index
k	the location/κ-subregion index
l	the frequency/sequency index
n	the number of κ-simplices in C_κ
σ, τ	a κ-simplex in a complex
α	the (κ 1)-simplex face of a κ-simplex in a complex
β	the $(\kappa+1)$-simplex hull of a κ-simplex in a complex
$L, L^{\mathrm{sym}}, L^{\mathrm{rw}}$	the (normalized) graph Laplacian
$L_\kappa, L_\kappa^{\mathrm{sym}}, L_\kappa^{\mathrm{rw}}$	the (normalized) Hodge κ-Laplacian
P	discrete orthogonal projection against the constant vector
B	the reciprocal-weight graph distance matrix
K, K^{sym}	the (normalized) harmonic kernel
H, H^{sym}	the (normalized) discrete integral operator
$\mathcal{L}$	the continuous Laplacian on some $\mathbb{R}^d$
$\mathcal{K}$	the continuous integral operator against the harmonic kernel
$\mathcal{P}$	continuous orthogonal projection against the constant function
G_k^j	the subgraph of G of scale j at location k
V_k^j	the vertices of G_k^j
C_k^j	the κ-subregion of C_κ of scale j at location k
$\mathbf{1}, \mathbf{1}_S$	constant vector, indicator function for a set S
$\boldsymbol{\phi}_i$	an eigenvector, usually of some Laplacian variant
$\boldsymbol{\phi}_{k,l}^j$	a κ-HGLET basis vector
$\boldsymbol{\psi}_{k,l}^j$	a κ-GHWT basis vector

The production of this dissertation involves a ton of notation, between wavelet indices, simplices of various order, and a variety of continuous and discrete, differential and integral operators. Vectors are always written in bold, like $\boldsymbol{x}$. Matrix and vector indexing is always written with square brackets, like $[M]_{ij}$ or $[\boldsymbol{x}]_i$. Some notation which is used consistently is collected in the above table.

Many of the figures and tables in this dissertation can be generated using the accompanying Julia package `MultiscaleSimplexSignalTransforms`, which we have released via the UC Davis TRIPODS Github organization at
`https://github.com/UCD4IDS/MultiscaleSimplexSignalTransforms.jl`.

We list these figures and tables below.

1.3. List of Figures and Tables

Figures

Tables

CH PTER 2

Background

2.1. Wavelets and Wavelet Packets

building block of real harmonic analysis is the Fourier transform, traditionally defined for a function $f \in L^1(\mathbb{R})$ via [**89**]

$$\mathcal{F}f(\xi) \coloneqq \widehat{f}(\xi) \coloneqq \int_{\mathbb{R}} f(t) \mathrm{e}^{\ -2\pi \mathrm{i}\xi t} \, \mathrm{d}t \ .$$

Considering f as a function over time, this transform allows us to *analyze* f in the frequency domain. When additionally $\widehat{f} \in L^1(\mathbb{R})$, then the inverse Fourier transform $\mathcal{F}^{\ -1} : L^1(\mathbb{R}) \mapsto L^1(\mathbb{R})$ recovers f, via a *synthesis* of the transform values [**89**]:

$$\left(\mathcal{F}^{\ -1} \widehat{f}\right)(t) \coloneqq \int_{\mathbb{R}} \widehat{f}(\xi) \mathrm{e}^{2\pi \mathrm{i}\xi t} \, \mathrm{d}\xi = f(t) \ .$$

density argument via compactly supported smooth functions is sufficient to extend the Fourier transform and inversion formula to $L^2(\mathbb{R})$ [**67**].

Intuitively, f being sufficiently smooth means it must vary slowly, which is equivalent to the transform values for higher frequencies being small; this is a very desirable property. For example, if $f \in C^p(\mathbb{R})$ and $f, \partial f, \ldots, \partial^p f \in L^1(\mathbb{R})$, then $\widehat{f}(\xi) \in \mathcal{O}(\xi^{\ -p})$. Conversely, if $\widehat{f} \in \mathcal{O}(\xi^{\ -(p+1+\epsilon)})$ for some $\epsilon > 0$, then $f \in C^p(\mathbb{R})$ [**89**].[1]

let $\mathbf{1}_S$ denote the indicator function for the set S. Notice that $\mathcal{F}(\mathrm{e}^{\ -|t|})\big)(\xi) \in \mathcal{O}(\xi^{\ -2})$, and $\mathcal{F}(\mathbf{1}_{[-1,1]})\big)(\xi) \in \mathcal{O}(\xi^{\ -1})$; these examples illustrate the typical phenomenon that the decay of $\widehat{f}$ is *globally* sensitive to the smoothness of f in time, so that a single discontinuity in f or a derivative of f affects the decay of the entire transform, which, e.g., increases the number of coefficients that may be needed to represent f to a given accuracy by discrete means [**56**].

[1] These are loose bounds to illustrate the connection.

Multiscale transforms aim to address this weakness by constructing transform values which are sensitive only *locally* in changes to smoothness, at different scales in both time and frequency, and exhibit decay properties appropriate to those local and transient phenomena. The *wavelet transform* is such an example, constructed from a family of translations and dilations of appropriate differencing (wavelet) and averaging (scaling) functions.

wavelet, or *mother wavelet* $\psi \in L^2(\mathbb{R})$ must satisfy $\|\psi\|_2 = 1$, and

$$\int_{\mathbb{R}} \psi(t)\,\mathrm{d}t = \widehat{\psi}(0) = 0 \ ,$$

so that fundamentally, it oscillates. *scaling function* ϕ, also known as a *father wavelet*, must satisfy $\|\phi\|_2 = 1$ **[67]**. With the translation operator defined as

$$T_u f(t) \coloneqq f(t \quad u) \ ,$$

and dilation operator defined for $s > 0$ as

$$D_s f(t) \coloneqq \frac{1}{\sqrt{s}} f\left(\frac{t}{s}\right) \ ,$$

the wavelet family is constructed from applications of dyadic dilation and translations to ψ, with scales $s = 2^j$ and time positions $u = 2^j k$ for $j, k \in \mathbb{Z}$. We define **[56]**

$$\phi_{j,k}(t) \coloneqq T_{2^j k} D_{2^j} \phi(t) = \frac{1}{\sqrt{2^j}} \phi\left(\frac{t \quad 2^j k}{2^j}\right) \ , \tag{2.1}$$

$$\psi_{j,k}(t) \coloneqq T_{2^j k} D_{2^j} \psi(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t \quad 2^j k}{2^j}\right) \ , \tag{2.2}$$

and the continuous wavelet transform of a function $f \in L^2(\mathbb{R})$ at scale $s = 2^j$ and time position $u = 2^j k$ as **[67]**

$$W f(j,k) \coloneqq \langle f, \psi_{j,k} \rangle = \int_{\mathbb{R}} f(t) \overline{\frac{1}{\sqrt{2^j}} \psi\left(\frac{t \quad 2^j k}{2^j}\right)}\,\mathrm{d}t \ . \tag{2.3}$$

Suppose the wavelets $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ form a *frame*, so that $\exists A, B > 0$ with $A \leq B$, such that

$$A\|f\|_2^2 \leq \sum_{j,k} |\langle f, \psi_{j,k} \rangle|^2 \leq B\|f\|_2^2 \quad \text{for } f \in L^2(\mathbb{R}) \ .^2$$

[2] When $A = B$, the wavelets form a *tight frame*.

Then there exist *dual wavelets* $\tilde{\psi}_{j,k}$ and *dual scaling functions* $\tilde{\phi}_{j,k}$, with which we can reconstruct f in the sense of L^2 [**56**]:

$$f(t) = \sum_{k\in\mathbb{Z}} \langle f, \phi_{j_{\max},k}\rangle \, \tilde{\phi}_{j_{\max},k}(t) + \sum_{k\in\mathbb{Z}} \sum_{j=\ -\infty}^{j_{\max}} \langle f, \psi_{j,k}\rangle \, \tilde{\psi}_{j,k}(t) \ . \tag{2.4}$$

These dual functions satisfy biorthogonality relations with the original ones [**67**]:

$$\begin{aligned} \left\langle \psi_{j,k}, \tilde{\psi}_{j',k'} \right\rangle &= \delta(j \quad j')\delta(k \quad k') && \text{for } j, j', k, k' \in \mathbb{Z} \ , \\ \left\langle \phi_{j,k}, \tilde{\phi}_{j,k'} \right\rangle &= \delta(k \quad k') && \text{for } j, k, k' \in \mathbb{Z} \ , \\ \left\langle \phi_{j,k}, \tilde{\psi}_{j,k'} \right\rangle &= \left\langle \psi_{j,k}, \tilde{\phi}_{j,k'} \right\rangle = 0 && \text{for } j, k, k' \in \mathbb{Z} \ , \end{aligned} \tag{2.5}$$

where δ is the usual Kronecker delta. We assume that ψ, ϕ satisfy both the frame condition, and further that each $\tilde{\psi}_{j,k} = \psi_{j,k}$, and $\tilde{\phi}_{j,k} = \phi_{j,k}$, so that ψ, ϕ form *orthogonal wavelets*. This allows for efficient implementation of both analysis and synthesis with the continuous wavelet transform, and for discrete-time signals. Define the spaces

$$\begin{aligned} \mathcal{V}_j &\coloneqq \overline{\operatorname{span}\left(\{\phi_{j,k}\}_{k\in\mathbb{Z}}\right)} \ , \\ \mathcal{W}_j &\coloneqq \overline{\operatorname{span}\left(\{\psi_{j,k}\}_{k\in\mathbb{Z}}\right)} \ . \end{aligned}$$

Then the biorthogonality relations (2.5) yield the orthogonal relations

$$\begin{aligned} \mathcal{W}_j \perp \mathcal{W}_{j'} \quad &\text{for } j \neq j' \ , \\ \mathcal{V}_j \perp \mathcal{W}_j \ , & \end{aligned}$$

and the spaces $\{\mathcal{V}_j\}_{j\in\mathbb{Z}}$ form a *multiresolution approximation*, described by the relations [**67**], [**25**]

$$\begin{aligned} f \in \mathcal{V}_j \iff D_2 f \in \mathcal{V}_{j+1} \ , \\ \mathcal{V}_j \supset \mathcal{V}_{j+1} \ , \\ \{0\} = \bigcap_{j\in\mathbb{Z}} \mathcal{V}_j \ , \\ L^2(\mathbb{R}) = \overline{\bigcup_{j\in\mathbb{Z}} \mathcal{V}_j} \ . \end{aligned}$$

Finally, because $\mathcal{V}_j = \mathcal{V}_{j+1} \oplus \mathcal{W}_{j+1}$, we can construct wavelets $\phi_{j,k}, \psi_{j,k}$ as linear combinations of finer-scale scaling functions:

$$\begin{aligned} \phi_{j,k}(t) &= \sum_{k' \in \mathbb{Z}} \underbrace{h(k' \quad 2k)}_{=\langle \phi_{j,k}, \phi_{j \quad 1,k'} \rangle} \phi_{j \quad 1,k'}(t) \ , \\ \psi_{j,k}(t) &= \sum_{k' \in \mathbb{Z}} \underbrace{g(k' \quad 2k)}_{=\langle \psi_{j,k}, \phi_{j \quad 1,k'} \rangle} \phi_{j \quad 1,k'}(t) \ , \end{aligned} \tag{2.6}$$

where h, g are called low-pass and high-pass filters respectively **[67]**, and are critically independent of j, allowing construction of the wavelet transform purely from repeated application of the filter relations (2.6), rather than explicitly constructing the wavelet family, and inner products against each function in it. Explicitly, the synthesis formula (2.4) becomes

$$f(t) = \sum_{k \in \mathbb{Z}} c_{j_{\max}}(k) \phi_{j_{\max},k}(t) + \sum_{k \in \mathbb{Z}} \sum_{j= \quad \infty}^{j_{\max}} d_j(k) \psi_{j,k}(t) \ , \tag{2.7}$$

where

$$\begin{aligned} c_j(k) &\coloneqq \langle f, \phi_{j,k} \rangle = \sum_{k' \in \mathbb{Z}} h(k' \quad 2k) c_{j \quad 1}(k') \ , \\ d_j(k) &\coloneqq \langle f, \psi_{j,k} \rangle = \sum_{k' \in \mathbb{Z}} g(k' \quad 2k) c_{j \quad 1}(k') \ . \end{aligned} \tag{2.8}$$

The c_j's and d_j's are called the *scaling coefficients* and *wavelet coefficients*, respectively.

Now we move to building the discrete wavelet transform, and consider a dyadic discrete signal $\boldsymbol{f} \in \mathbb{R}^N$, where $N = 2^{n_0}$, $n_0 \in \mathbb{Z}_{\geq 0}$. Taking $j = 0$ to be the finest level, and $j = j_{\max}$ to be the coarsest with $j_{\max} \leq n_0$, we define the finest scaling coefficients to be just the signal itself, so that $c_0(k) \coloneqq [\boldsymbol{f}]_k$, $k = 1, \ldots, N$. Using (2.6), we can then obtain the c_j, d_j coefficients for $j > 0$; because of the finite number of scales j and time positions k, the formula simplifies to

$$[\boldsymbol{f}]_n = \sum_{k=0}^{2^{n_0} \quad j_{\max} \quad 1} c_{j_{\max}}(k) \phi_{j_{\max},k}(n) + \sum_{k=0}^{2^{n_0} \quad j \quad 1} \sum_{j=1}^{j_{\max}} d_j(k) \psi_{j,k}(n) \ . \tag{2.9}$$

The recursive structure of the coefficient calculation lends itself to a *Fast Wavelet Transform* analogous to the Fast Fourier Transform (FFT), only with a faster time cost of $\mathcal{O}(N)$ operations, rather than $\mathcal{O}(N \log N)$ **[21]**.

The wavelet transform succeeds at identifying features of signals at various time-scales; however, the resolution of the transform is limited, in that there is an explicit inverse relationship between the time-localization and frequency of the corresponding wavelet functions. For example, wavelet transforms provide poor representations for high-frequency components, which an ordinary or windowed Fourier transform might capture well. This is due to the construction, in which filters are successively applied only to scaling coefficients.

The *wavelet packets* of Coifman, Meyer, and Wickerhauser [**19**] address the resolution limitations by applying low- and high-pass filters to the finer-scale (i.e., higher-frequency) wavelet functions as well. Following the notation in [**95**], the level parameter l is introduced alongside scale j and time position k. The scaling functions occupy $l = 0$, the wavelet functions occupy $l = 1$, and higher-level wavelet packets for a given j, k are higher-frequency oscillations over the same region. The orthogonal wavelet packet functions are initialized with $w^0_{j,k}(t) \coloneqq \phi_{j,k}(t)$ and $w^1_{j,k}(t) = \psi_{j,k}(t)$, and then further generated according to

$$\begin{aligned} w^{2l}_{j,k} &\coloneqq \sum_{k'} h(k' \quad 2k) w^l_{j \;\; 1,k'}(t) \;, \\ w^{2l+1}_{j,k} &\coloneqq \sum_{k'} g(k' \quad 2k) w^l_{j \;\; 1,k'}(t) \;. \end{aligned} \tag{2.10}$$

Wavelet packet coefficients are initialized with $d^0_j(k) \coloneqq c_j(k)$ and $d^1_j(k) \coloneqq d_j(k)$, and then further generated via

$$\begin{aligned} d^{2l}_j(k) &\coloneqq \left\langle f, w^{2l}_{j,k} \right\rangle = \sum_{k'} h(k' \quad 2k) d^l_{j \;\; 1}(k') \;, \\ d^{2l+1}_j(k) &\coloneqq \left\langle f, w^{2l+1}_{j,k} \right\rangle = \sum_{k'} g(k' \quad 2k) d^l_{j \;\; 1}(k') \;. \end{aligned} \tag{2.11}$$

Returning to the discrete setting, with $\boldsymbol{f} \in \mathbb{R}^N$ and h, g as before, the wavelet packet coefficients can still be generated via (2.11), and thus produce an $N \times (n_0 + 1)$ matrix of transform coefficients; N coefficients for each of the n_0 scales $j > 0$, and the N original function values. The associated time cost for the discrete transform is also $N \log N = N n_0$.

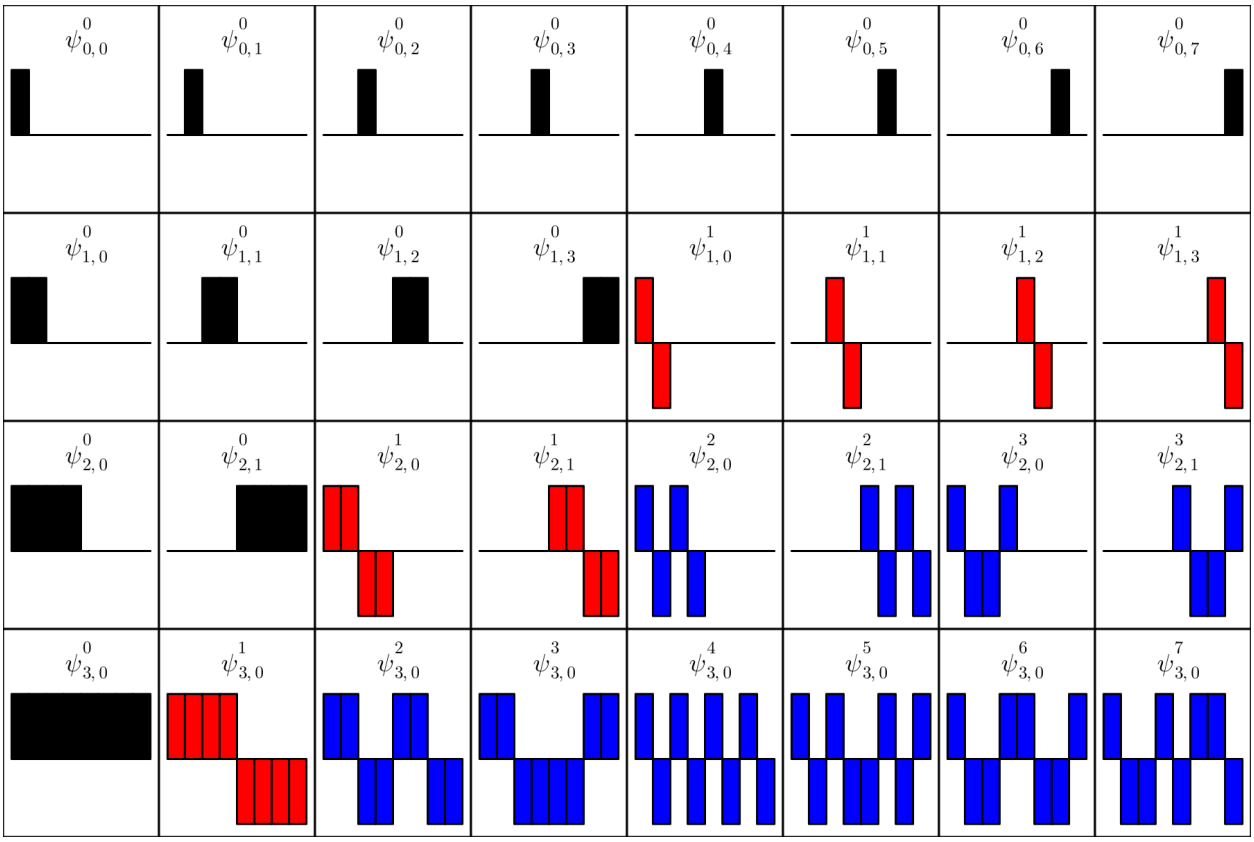


FIGURE 2.1. The Haar-Walsh wavelet packets $\boldsymbol{\psi}^l_{j,k}$ for a dyadic discrete signal on $\mathbb{R}^8$. Each row is at constant scale j, and displayed in blocks of constant l, and varying time position k. The scaling functions ($l = 0$) are in black, wavelet functions ($l = 1$) are in red, and wavelet packet functions ($l \geq 2$) are in blue. The bottom row consists of the Walsh functions. Figure design is a recreation of [**45**, Figure 2.4].

The *Haar-Walsh* wavelet packets are a classical example [**18**], which this dissertation revisits in the context of simplicial complexes. These packets are piecewise-constant, with mother wavelet

$$\psi(t) = \mathbf{1}_{[0,\frac{1}{2})} \quad \mathbf{1}_{[\ \frac{1}{2},0)} \ ,$$

and scaling function

$$\phi(t) = \mathbf{1}_{[\ \frac{1}{2},\frac{1}{2})} \ .$$

Figure 2.1 displays each of the Haar-Walsh wavelet packet functions for $N = 8$. The Haar scaling family ($l = 0$) consists of local bumps, the Haar wavelet family ($l = 1$) consists of single

local oscillations, and the higher-order Haar-Walsh wavelet packets consist of multiple oscillations, at possibly varying scales. The global oscillations in the bottom row of Figure 2.1 are known as the *Walsh functions*, and take only the values $\{\pm N^{-1/2}\}$. Rescaled to instead take values in $\{\pm 1\}$, the Walsh function gathered as column vectors form the $N \times N$ *Hadamard matrix* H_N, with the property that both its rows and columns are orthogonal, satisfying $H_N H_N = H_N H_N = N I_N$, with I_N the $N \times N$ identity matrix [**2**].

2.2. Graph Theory

Now, we establish the standard setting for spectral graph theory, and introduce the notation we will use.

Let $G = (V, E)$ be an undirected connected graph. $V = V(G) = \{v_1, \ldots, v_N\}$ is the vertex set of the graph, where $|V(G)| = N$. $E = E(G) = \{e_1, \ldots, e_M\}$ is the edge set of the graph, where $|E(G)| = M$. Each edge $e_l = \{v_{i_l}, v_{j_l}\}$ is a set of two vertices, indicating v_{i_l}, v_{j_l} are connected in the graph. We only consider *simple* graphs, which have no loops or multiple-edges, so that the two vertices making up each edge are distinct, and each edge appears in E at most once. For simplicity, when it is clear from context, we will write v_i as i. In a *directed* graph , each edge e is instead a tuple $(v_{i_l}, v_{j_l}) \neq (v_{j_l}, v_{i_l})$; the first vertex is the tail, and the second is the head.[3] *graph signal* on G is a function $\boldsymbol{f} : V \mapsto \mathbb{R}$, which we generally write with the same notation as a vector in $\mathbb{R}^N$, so that $[\boldsymbol{f}]_i = \boldsymbol{f}(v_i)$. Let $\mathbf{1}_S$ be the indicator function for a set $S \subset V$, so $[\mathbf{1}_S]_i = \delta(v_i \in S)$, and $\mathbf{1} \coloneqq \mathbf{1}_V$.

The structure of a graph G is captured by its adjacency matrix $W(G) \in \mathbb{R}^{N \times N}$, satisfying $[W]_{ij} = \delta(\{v_i, v_j\} \in E)$, so two vertices are adjacent when an edge connects them. *weighted* graph more generally has edge weight $[W]_{ij} > 0$ for each adjacent pair i, j of vertices, which indicate the *affinity* of those vertices, or the strength of the relationship between them. When G is undirected, W is a symmetric matrix. Generally, from now on we assume the underlying graph G is undirected, and otherwise will refer to the directed graph .

s an aside, when constructing a graph from scattered data, edge weights between data must meaningfully capture their affinity. For example, if the vertices represents discrete samples of a

[3] In Section 2.4, we will re-interpret a directed edge instead as an oriented simplex.

manifold, then the weight between samples could depend *inversely* on the geodesic distance between them. In **[8]**, the celebrated correspondence between the eigenfunctions of the Laplace-Beltrami operator of a manifold, and the eigenvectors of L^{rw} for a specially-constructed graph on points sampled from that manifold, can be established by choosing the edge weights between adjacent points $x_i, x_j \in \mathbb{R}^d$, $d \in \mathbb{Z}_{>0}$ as $\exp(\quad \|x_i \quad x_j\|^2/t)$ for some scale parameter $t > 0$.

Returning to G, let the degree of v_i be $[\boldsymbol{d}]_i \coloneqq \sum_j [W]_{ij}$, and define the *degree matrix*

$$D(G) \coloneqq \operatorname{diag}(\boldsymbol{d}) \ .$$

The key operator we will examine is the *combinatorial Laplacian* matrix, together with its most common variations, the *random-walk normalized Laplacian* and *symmetric normalized Laplacian*, defined respectively as

$$\begin{aligned} L(G) &\coloneqq D(G) \quad W(G) \ , \\ L^{\text{rw}}(G) &\coloneqq D(G)^{\ 1} L(G) \ , \\ L^{\text{sym}}(G) &\coloneqq D(G)^{\ 1/2} L(G) D(G)^{\ 1/2} \ . \end{aligned}$$

For any of these Laplacian variations, $0 = \lambda_0 \leq \lambda_1 \leq \ldots \lambda_{N\ 1}$ will refer to the sorted Laplacian eigenvalues, and $\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{N\ 1}$ will refer to the corresponding Laplacian eigenvectors. If the context is unclear, superscripts (e.g., $\boldsymbol{\phi}_1^{\text{rw}}$) or explicit reference (e.g., $\boldsymbol{\phi}_1(L^{\text{rw}})$) will indicate the referred Laplacian variation.

The spectral properties of the Laplacians are well-studied and celebrated, and we'll describe just those that best develop spectral clustering and partitioning; for the rest see **[16, 96]**. First, the quadratic forms associated with L and L^{sym} are

$$\boldsymbol{f}^{\ } L\boldsymbol{f} = \frac{1}{2} \sum_{ij} [W]_{ij} \left([\boldsymbol{f}]_i \quad [\boldsymbol{f}]_j \right)^2 \ , \tag{2.12}$$

$$\boldsymbol{f}^{\ } L^{\text{sym}} \boldsymbol{f} = \frac{1}{2} \sum_{ij} [W]_{ij} \left(\frac{[\boldsymbol{f}]_i}{\sqrt{[\boldsymbol{d}]_i}} \quad \frac{[\boldsymbol{f}]_j}{\sqrt{[\boldsymbol{d}]_j}} \right)^2 \ . \tag{2.13}$$

Hence the bottom eigenpairs $(\lambda_0, \boldsymbol{\phi}_0)$ of the symmetric matrices L, L^{sym} are $(0, \mathbf{1})$, $(0, D^{1/2}\mathbf{1})$ respectively, so L, L^{sym} are positive semidefinite, and the rest of the eigenvalues are positive in

both cases exactly when G is connected. Next, L^{rw} and L^{sym} have the same eigenvalues, and their eigenvectors are related by

$$\boldsymbol{\phi}_i^{\text{rw}} = D^{\ 1/2}\boldsymbol{\phi}_i^{\text{sym}} \ . \tag{2.14}$$

s L^{rw} is not symmetric, its eigenvectors need not be orthonormal; instead they are orthonormal in the degree-weighted inner product:

$$(\boldsymbol{\phi}_i^{\text{rw}})^{\ } \ D\boldsymbol{\phi}_j^{\text{rw}} = \delta_{ij} \ . \tag{2.15}$$

Finally, Discrete Nodal Domain Theorems **[26]** demonstrate that the sign of $\boldsymbol{\phi}_k$ partitions V such that the resulting induced subgraphs are connected, with higher eigenvalue index tending towards more of and finer such subgraphs. In particular, Fiedler made the foundational discovery that an eigenvector corresponding to the first nonzero eigenvalue of L bipartitions V such that the two induced subgraphs are connected **[31]**. For a connected graph, $\boldsymbol{\phi}_1$ is famously called the *Fiedler vector*, and in many analogous settings, when the assignment of a bipartition of some set is derived from top or bottom eigenvectors of an operator on that set, it is common to refer to that assignment (and/or a relevant eigenvector) as a Fiedler vector. We will continue this tradition. In this dissertation, we focus on hierarchical bipartitioning, but we will mention that the methods used to derive the Fiedler vector for bipartitioning with Laplacian variations generally extend to k-way clustering by treating $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_k$ as a k-dimensional *embedding* of the N vertices, and then using a standard technique in $\mathbb{R}^k$, such as k-means or some variation of it **[27, 96]**. We will briefly revisit embeddings in Chapter 3.

Both facts (2.14, 2.15) are central in justifying the use of the eigenvectors of L^{sym} to partition graphs, as shown in **[87, 96]**. The vertex partition induced by the sign of the Fiedler vector derived from each Laplacian variation solves a relaxed version of an NP-hard combinatorial optimization graph-cut problem. For each of $L, L^{\text{rw}}, L^{\text{sym}}$, that optimization objective is $\min_S \text{Cut}(S, S^c)$,

$\min_S \mathrm{RatioCut}(S, S^c)$, and $\min_S \mathrm{NormalizedCut}(S, S^c)$ respectively, each defined by

$$\begin{aligned}
\mathrm{Cut}(A, B) &\coloneqq \sum_{i \in A, j \in B} [W]_{ij} \ , \\
\mathrm{Vol}(A) &\coloneqq \mathrm{Cut}(A, A) \ , \\
\mathrm{RatioCut}(A, B) &\coloneqq \mathrm{Cut}(A, B) \left(\frac{1}{|A|} + \frac{1}{|B|} \right) \ , \\
\mathrm{NormalizedCut}(A, B) &\coloneqq \mathrm{Cut}(A, B) \left(\frac{1}{\mathrm{Vol}(A)} + \frac{1}{\mathrm{Vol}(B)} \right) \ .
\end{aligned} \tag{2.16}$$

Observe that

$$\sum_{ij} [W]_{ij} = \mathrm{Vol}(S) + \mathrm{Vol}(S^c) + 2\,\mathrm{Cut}(S, S^c) \ ,$$

and is independent of S, so the goal of bipartition can be thought of equivalently as either finding a bipartition which minimizes edge weight crossing between partitions, or finding one which maximizes volume contained on each side of the partition, in both subject to some normalization; $|S|$ in the case of Ratio Cut, and $\mathrm{Vol}(S)$ in the case of Normalized Cut.

Now, one of the fundamental differences when we consider simplicial complexes will be *orientation*, naturally leading to the consideration of *signed graphs*. signed graph has the form $G = (V, E^{\pm})$, where $[W(G)]_{ij} > 0$ when $\{i, j\} \in E^+$, and $[W(G)]_{ij} < 0$ when $\{i, j\} \in E$ (and necessarily, $E^+ \cap E = \emptyset$). While $L(G)$ may then no longer be positive semidefinite, some work has shown that eigenvectors corresponding to negative eigenvalues of L may still be used for clustering **[58]**. However, critically, the nodal domain theorem may not hold, so more advanced interpretation is required.

The *signed Laplacian* framework **[62]** instead recovers desirable properties of L for signed graphs, by using the absolute weights $\left[\overline{W}(G)\right]_{ij} = |[W(G)]_{ij}|$ and absolute degrees

$$[\overline{\boldsymbol{d}}]_i \coloneqq \sum_j \left[\overline{W}(G)\right]_{ij} \qquad , \qquad \overline{D}(G) \coloneqq \mathrm{diag}(\overline{\boldsymbol{d}}) \ ,$$

to construct the signed Laplacian $\overline{L}(G) \coloneqq \overline{D}(G) \quad W(G)$. The variants $\overline{L}^{\mathrm{sym}}, \overline{L}^{\mathrm{rw}}$ are constructed identically, using $\overline{D}$ in place of D. Key for our purposes, from **[62]** we know that analogous to

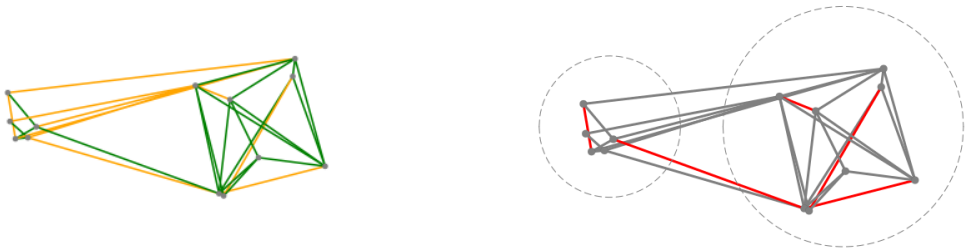


FIGURE 2.2. On the left, a simple example of a signed graph, with vertices in grey, positive edges in green, and negative edges in orange. On the right, a bipartition of the same signed graph, where the edges which contribute to the SignedCut objective are highlighted in red.

equations (2.12, 2.13), we have

$$\boldsymbol{f}^{\top}\overline{L}\boldsymbol{f} = \frac{1}{2}\sum_{ij}[\overline{W}]_{ij}\left([\boldsymbol{f}]_i - \operatorname{sgn}([W]_{ij})[\boldsymbol{f}]_j\right)^2 \geq 0 \quad \text{, and} \tag{2.17}$$

$$\boldsymbol{f}^{\top}\overline{L}^{\text{sym}}\boldsymbol{f} = \frac{1}{2}\sum_{ij}[\overline{W}]_{ij}\left(\frac{[\boldsymbol{f}]_i}{\sqrt{[\overline{\boldsymbol{d}}]_i}} - \operatorname{sgn}([W]_{ij})\frac{[\boldsymbol{f}]_j}{\sqrt{[\overline{\boldsymbol{d}}]_j}}\right)^2 \geq 0\ . \tag{2.18}$$

For each of $\overline{L}, \overline{L}^{\text{rw}}, \overline{L}^{\text{sym}}$, the vertex partition induced by the sign of the corresponding Fiedler vectors $\boldsymbol{\phi}_1(\overline{L}), \boldsymbol{\phi}_1(\overline{L}^{\text{rw}}), \boldsymbol{\phi}_1(\overline{L}^{\text{sym}})$ solve a relaxed version of the combinatorial optimizations $\min_S \text{SignedCut}(S, S^c)$, $\min_S \text{SignedRatioCut}(S, S^c)$, $\min_S \text{SignedNormalizedCut}(S, S^c)$, respectively, each defined by

$$\begin{aligned}
\text{Cut}^{\pm}(A, B) &\coloneqq \sum_{i\in A, j\in B} \max(0, \pm[W]_{ij})\ , \\
\text{Vol}^{\pm}(A) \coloneqq \text{Cut}^{\pm}(A, A) \quad &, \quad \text{AbsVol}(A) \coloneqq \text{Vol}^{+}(A) + \text{Vol}^{-}(A)\ , \\
\text{SignedCut}(A, B) &\coloneqq 2\,\text{Cut}^{+}(A, B) + \text{Vol}^{-}(A) + \text{Vol}^{-}(B)\ , \\
\text{SignedRatioCut}(A, B) &\coloneqq \text{SignedCut}(A, B)\left(\frac{1}{|A|} + \frac{1}{|B|}\right)\ , \\
\text{SignedNormalizedCut}(A, B) &\coloneqq \text{SignedCut}(A, B)\left(\frac{1}{\text{AbsVol}(A)} + \frac{1}{\text{AbsVol}(B)}\right)\ .
\end{aligned}$$

The goal of this bipartition is to simultaneously minimize *positive* edge weight crossing between partitions and *negative* edge weight contained within either side of the partition, such as those of

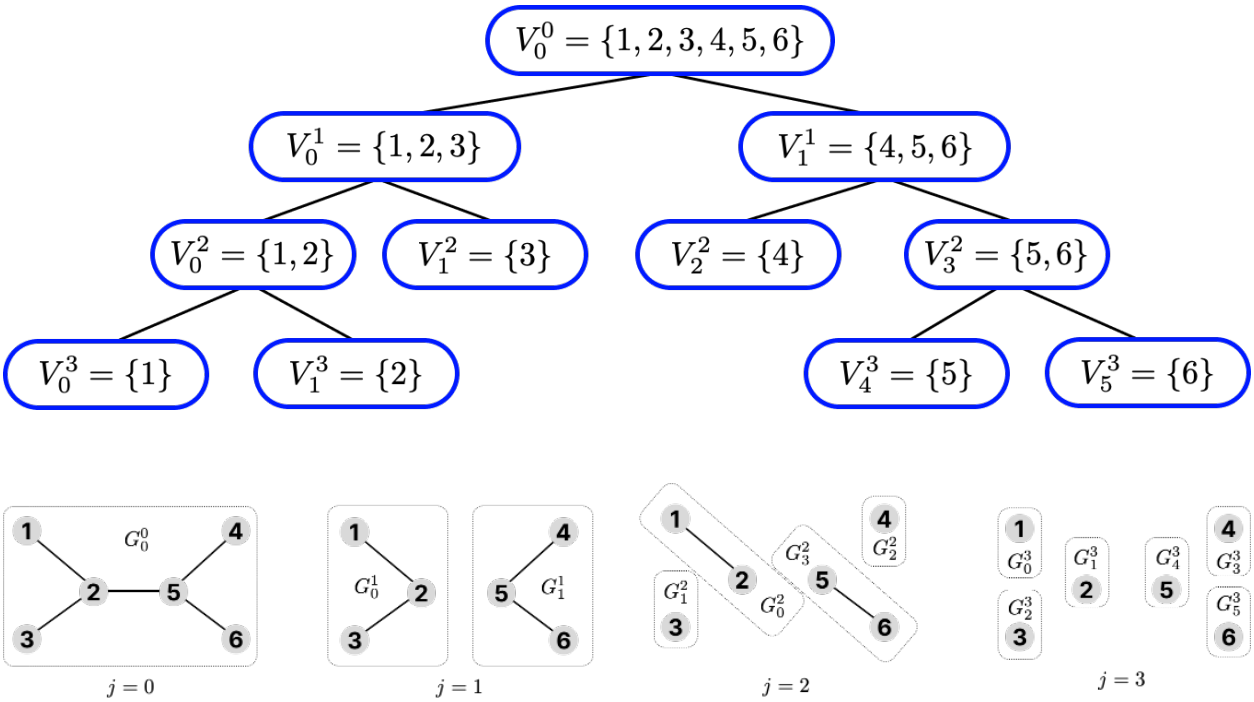


FIGURE 2.3. hierarchical partition tree, for the 6 vertices of a simple example graph. In the tree above, the root is V_0^0, containing all the vertices, and each next level of the tree contains subsets at common scale. The leaves of the tree are singleton subsets. In the graphs below, the regions G_k^j induced by the corresponding vertex subsets V_k^j are demonstrated.

the red edges in Figure 2.2. Equivalently, we seek to simultaneously *maximize* positive edge weight contained within either side of the partition and negative edge weight crossing between partitions, so that as much as possible, positive edges indicate affinity, and negative edges indicate repulsion.

Spectral clustering of signed networks remains an active field of research, and the discovery of new methods such as the Signed Positive Over Negative Generalized Eigenproblem (SPONGE) algorithm [**24**] has shown rigorous guarantees for identifying planted clusters in signed stochastic block models (SSBMs) for the Signed Laplacian, and additionally that SPONGE clustering enjoys such guarantees under a greater variety of conditions.

2.3. Recursive Graph Partitioning

The foundation on which the multiscale transforms studied in this dissertation are constructed is *hierarchical bipartition trees* or *binary partition trees* of the underlying sets.

Such a tree organizes subsets of the underlying set into clusters at various scales, an example of which can be seen in Figure 2.3. In the case of a graph $G = (V, E)$, the underlying set is V, and we denote the subsets by $V_k^j \subseteq V =: V_0^0$, with j denoting the scale index, and k denoting a position index, indexing the sets at a given scale. Increasing j goes from coarse scale to fine scale, with $j = 0$ containing only V, and $j = j_{\max}$ containing singleton $\{v_i\}$'s. We define G_k^j as the subgraph of G induced by restricting it to the vertices in V_k^j. The relationship between G_k^j and V_k^j is illustrated in Figure 2.3. Each such subgraph G_k^j will be referred to as a *region*, as an analogy to the localized regions on which wavelets concentrate. When a region is contained in another region, we may refer to it as a *subregion*. Given a graph, we will typically form a hierarchical partition tree top-down, using the information in each region G_k^j to bipartition the vertices V_k^j into nonempty, disjoint child subsets $V_{k'}^{j+1}, V_{k'+1}^{j+1}$, forming the subregions $G_{k'}^{j+1}, G_{k'+1}^{j+1}$. Our standard method will be to use the sign of the Fielder vector for an appropriate Laplacian variant to perform the bipartition at each stage; the critical property is that the vector oscillates, so that neither partition is empty. bonus property is that $|V_{k'}^{j+1}|, |V_{k'+1}^{j+1}|$ should be close in size, which, e.g., L^{rw}-based partitions satisfy due to their proximity to NormalizedCut optimizers. In general however, any bipartitioning method which generates nonempty disjoint partitions may be used to construct a top-down hierarchical partition tree. n example of the partitions formed for a synthetic graph is provided in Figure 2.4.

In previous work involving L^{rw}-based hierarchical partitioning (e.g., [**47**,**48**]), the matrices L^{sym} are explicitly re-computed[4] for each G_k^j, and the relevant Fiedler vectors then calculated. In this dissertation, we will also examine a second approach to hierarchical partitioning we call *submatrix partitioning*, which takes advantage of the fact that the Laplacian for G_k^j has a close relationship with the submatrix of the Laplacian for G_0^0, induced by V_k^j. Simply, instead of re-computing L^{sym} at each stage of the hierarchical partition, we approximate it using the appropriate submatrix of the original Laplacian. We pursue this approximation for two reasons. First, while the submatrix relationship is straightforward for hierarchical partitioning of vertices in a graph, as we will introduce next in Section 2.4 and then explain in detail in Chapter 4, performing hierarchical partitions over the analogous higher-dimensional structures (e.g., κ-regions consisting of κ-simplices, with $\kappa \geq 1$) poses new difficulties worth studying, that can be overcome through approximation. Second, when

[4]Even when we desire the eigenvectors of L^{rw}, typically we take advantage of symmetric eigenvector methods using L^{sym}, and then transform the results as needed.

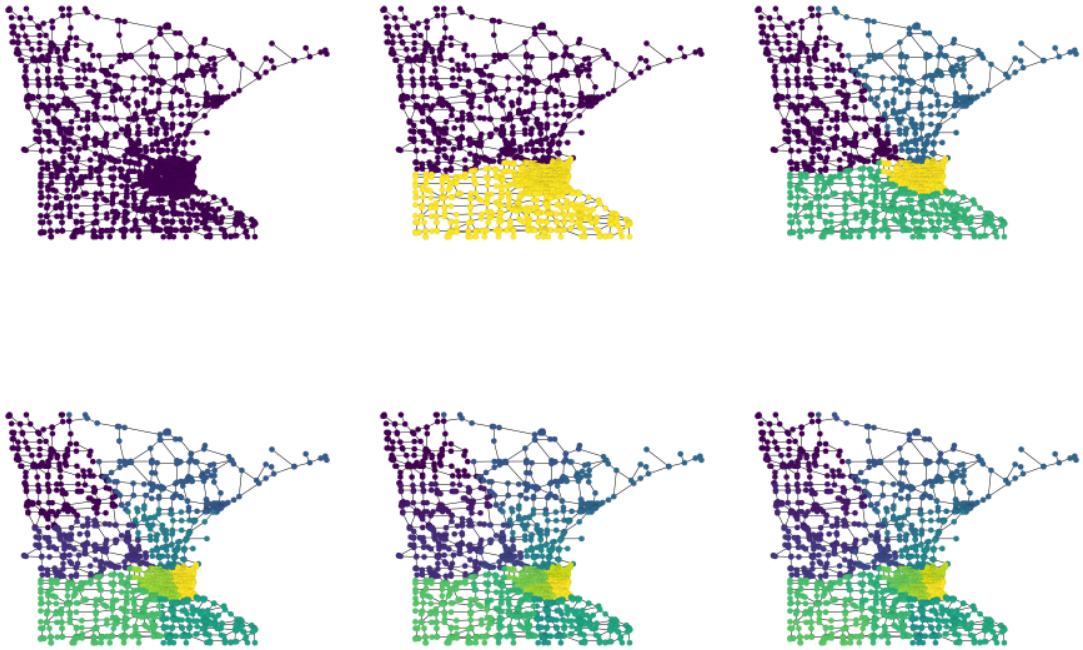


FIGURE 2.4. n example of a hierarchical partitioning of a graph, using a Minnesota road network **[36]**. Each plot, from left to right, top to bottom, includes one further level of the partition. Vertices which share the same color, belong to the same level of the partition, and vertices nearer in color to each other have a nearer common ancestor in the hierarchical partition tree.

the approximation is justified, the computation of a hierarchical partition can be dramatically sped up, and be made more stable, by either avoiding the re-computation of Laplacians altogether, or developing data structures which minimize the need for re-computation.

Unless stated otherwise, going forward, submatrix partitioning is the method we use for experiments and calculations when performing hierarchical partitioning via Fiedler vectors, whether of L or L^{sym}.

Now, we first analyze the symmetric normalization. Let W_{sub} denote the $|V_k^j| \times |V_k^j|$ matrix obtained by restricting the rows and columns of $W(G_0^0)$ to indices in V_k^j. Similarly, use the same subscript to denote all of the analogous $|V_k^j| \times |V_k^j|$ submatrix quantities D_{sub}, $W_{\text{sub}}^{\text{sym}}$, L_{sub}, $L_{\text{sub}}^{\text{sym}}$. Let W_{jk} denote the usual $|V_k^j| \times |V_k^j|$ adjacency matrix $W(G_k^j)$, and use the same subscript to

denote the analogous quantities D_{jk}, W^{sym}_{jk}, L_{jk}, L^{sym}_{jk}. Let's consider a vertex $v_i \in V^j_k$. Notice that $[D_{\text{sub}}]_{ii} \geq [D_{jk}]_{ii}$, because G^j_k is a subgraph of G^0_0. Let $[\boldsymbol{f}]_i \coloneqq ([D_{\text{sub}}]_{ii} \quad [D_{jk}]_{ii})/[D_{jk}]_{ii})$ be the relative difference in degrees, satisfying $[\boldsymbol{f}]_i \geq 0$, and let $F = \text{diag}(\boldsymbol{f})$. $\quad$ s $[L^{\text{sym}}_{\text{sub}}]_{ii} = [L^{\text{sym}}_{jk}]_{ii} = 1$, the diagonals are equal. Further, because we have $W_{\text{sub}} = W_{jk}$,

$$\begin{aligned} L^{\text{sym}}_{\text{sub}} = I \quad W^{\text{sym}}_{\text{sub}} &= I \quad (D_{jk} + (D_{\text{sub}} \quad D_{jk})) \ ^{\frac{1}{2}}(W_{jk})(D_{jk} + (D_{\text{sub}} \quad D_{jk})) \ ^{\frac{1}{2}} \\ &= I \quad (I + F) \ ^{\frac{1}{2}}W^{\text{sym}}_{jk}(I + F) \ ^{\frac{1}{2}} \\ &= (I + F) \ ^{\frac{1}{2}} \left(L^{\text{sym}}_{jk} + F\right) (I + F) \ ^{\frac{1}{2}} . \end{aligned}$$

Thus, the restricted $L^{\text{sym}}_{\text{sub}}$, acting as a perturbation of L^{sym}_{jk}, is simply a *more* diagonally dominant version of L^{sym}_{jk}. To get a feeling for the extreme scenarios, when $F = \epsilon I$, then $L^{\text{sym}}_{\text{sub}} = L^{\text{sym}}_{jk} + \frac{\epsilon}{1+\epsilon}W^{\text{sym}}_{jk}$, and when $F = (\epsilon \ ^1 \quad 1)I$, then $L^{\text{sym}}_{\text{sub}} = I \quad \epsilon W^{\text{sym}}_{jk}$. Now, recall that each successive partition of G solves a relaxed graph cut problem; that means we should expect F to increase slowly, on average, for each bipartition.

If $\|I + F\|^2 = \sum_i([\boldsymbol{f}]_i + 1)^2$ is sufficiently small, because we are dealing with real symmetric positive semidefinite matrices, and the perturbation is a diagonal matrix, we obtain the strong result that the submatrix Fiedler vector is itself a perturbation of the full-partition Fiedler vector [**43**]. Under these assumptions, the regions constructed by the submatrix partition will start identical at top level (i.e., the first bipartition), remain nearly identical so long as the successive partitions remove a small fraction of the total within-partition edges, and then degrade to more random partitions as region size shrinks to individual vertices, and the embedding becomes noisy eigenvectors of a perturbation of I. By a nearly identical argument to that of L^{sym}, we make the same conclusion for L. Figure 2.5 demonstrates this graceful decay of the faithfulness of the regions when so approximated.

2.4. Simplicial Complexes

In this section we review concepts from algebraic topology to formally define simplicial complexes and lift relevant notions from graph theory like adjacency to them. Our goal when we introduce new or non-standard language or notation (like *natural parity* below) is to be more precise and careful where we have found standard materials less intuitive. For a more thorough review see [**13**, **35**].

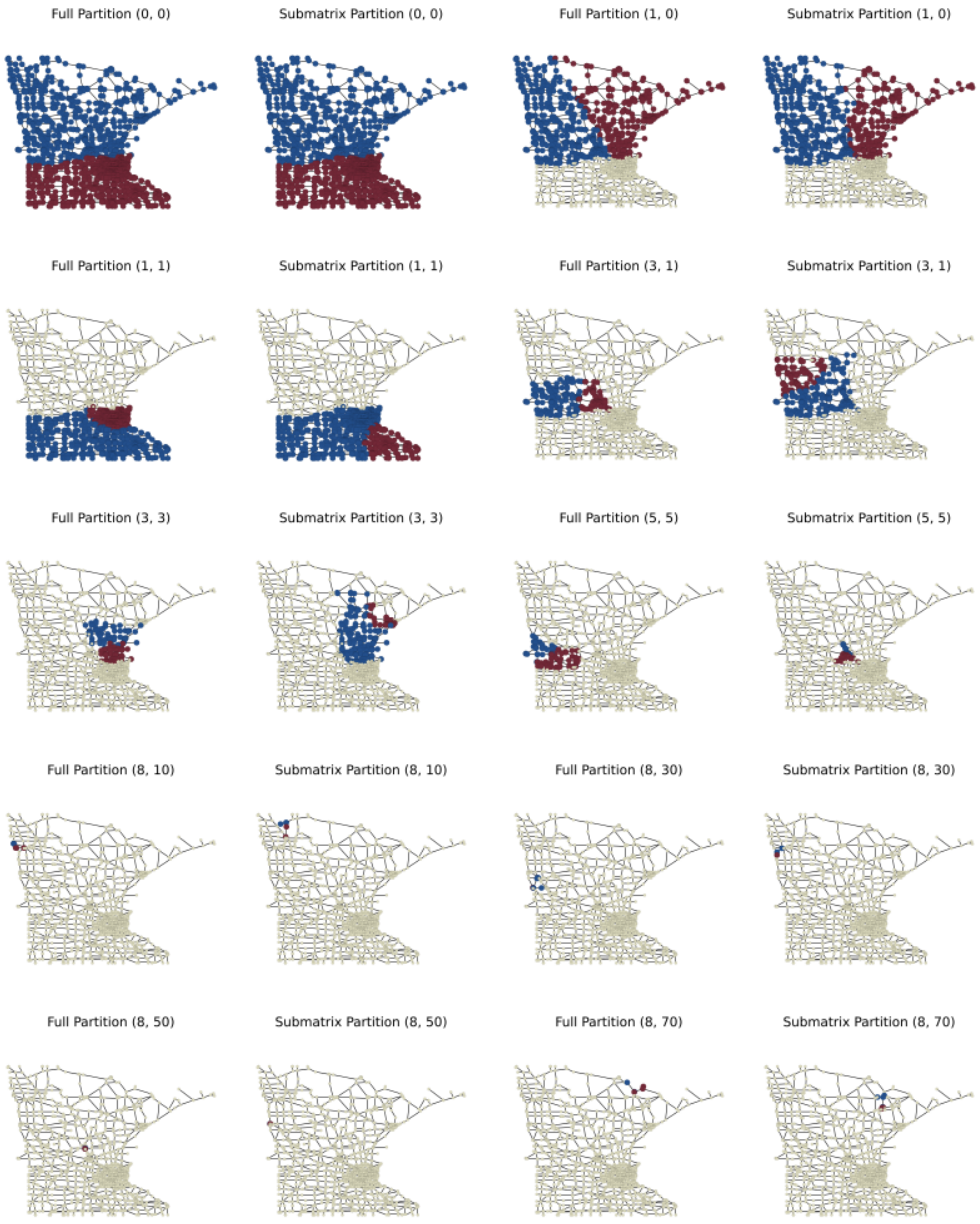


FIGURE 2.5. Several instances of an L-based hierarchical partitioning of a Minnesota road network **[36]**, with the results of usual full partitioning on the left in each pair, and those of submatrix partitioning on the right. The blue and red-colored vertices in each graph are the support of that region. The two indices in each label are the scale (j) and location (k) parameters, and the two uniformly colored subregions indicate the next partition for that region. Notice how the regions identified by both methods are similar, and that at later scale parameters, the much smaller supports vary more between the two methods.

Given a vertex set $V = \{v_1, \ldots, v_n\}$, a *κ-simplex* σ is a $(\kappa+1)$-subset of V. s we will also later define oriented simplices, for clarity let $V(\sigma)$ denote the vertices in any simplex σ. *face* of σ is a κ-subset of σ, and so σ has $\kappa+1$ faces. *co-face* of σ is a $(\kappa+1)$-simplex, of which σ is a face.

Suppose $\sigma = \{v_{i_1}, \ldots, v_{i_{\kappa+1}}\}$, $i_1 < \cdots < i_{\kappa+1}$, and $\alpha \subset \sigma$ is its face. Then, $|\sigma \setminus \alpha| = 1$. Let $\iota(\sigma \setminus \alpha)$ refer to the position of this single vertex in σ, such that

$$\sigma \setminus \alpha = \left\{ v_{i_{\iota\ \sigma\setminus\ )}} \right\} . \tag{2.19}$$

Define the *natural parity* of σ with respect to its face α as

$$\mathrm{nat}(\sigma, \alpha) := (\ \ 1)^{\iota(\sigma\setminus\ )+1} . \tag{2.20}$$

When α is not a face of σ, $\mathrm{nat}(\sigma, \alpha) = 0$. The natural parity of κ-simplices with respect to their faces generalizes the idea of a directed edge having a head vertex and a tail vertex, and is "natural" because it disallows situations analogous to a directed edge with two heads or two tails. In existing literature, this choice of parity is often taken as an assumption, or developed as a consequence of defining orientation or boundary maps with respect to permutation parity, as in **[14**, ppendix C].

Consider the case $\kappa = 1$. 1-simplex $e = \{v_i, v_j\}$, $i < j$, is equivalent to a directed edge from v_i to v_j. The two faces of e are the singletons $\{v_i\}$ and $\{v_j\}$, and notice $\mathrm{nat}(e, \{v_i\}) = \ \ 1$, and $\mathrm{nat}(e, \{v_j\}) = 1$. Usually v_j is called the head of the directed edge, and v_i the tail. For a κ-simplex σ, let $\sigma^{\mathrm{nh}} := \{\alpha \text{ is a face of } \sigma \mid \mathrm{nat}(\sigma, \alpha) > 0\}$ be the *natural heads* of σ, and likewise let $\sigma^{\mathrm{nt}} := \{\alpha \text{ is a face of } \sigma \mid \mathrm{nat}(\sigma, \alpha) < 0\}$ be the *natural tails* of σ. Then, $\{v_j\}$ is the natural head of e, and $\{v_i\}$ is the natural tail of e.

From the definition of natural parity, it is clear that for any κ-simplex σ, as σ has $\kappa+1$ faces, if κ is odd, then $|\sigma^{\mathrm{nh}}| = |\sigma^{\mathrm{nt}}| = \frac{\kappa+1}{2}$, and if κ is even, then $\big||\sigma^{\mathrm{nh}}| \ \ |\sigma^{\mathrm{nt}}|\big| = 1$. In other words, the faces of σ are as evenly as possible partitioned between the natural heads and natural tails; we say the natural head and natural tail sets are *balanced*.

simplicial complex C is a collection of simplices closed under subsets, where if $\sigma \in C$, then $\alpha \subset \sigma \implies \alpha \in C$. In particular, if $\sigma \in C$, so does each face of σ. Let $\kappa_{\max}(C) := \max\{\kappa \,|\, \sigma \in C \text{ is a } \kappa\text{-simplex}\}$, and let C_κ denote the set of κ-simplices in C for each $\kappa = 0, \ldots, \kappa_{\max}$.

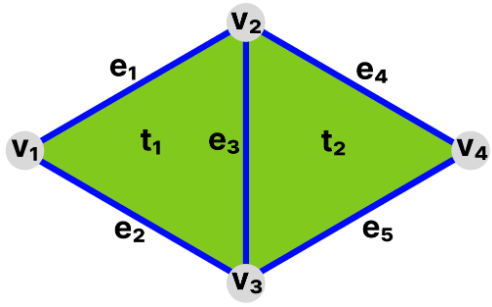


FIGURE 2.6. In this small 2-complex C, $e_1 \sim e_4$ because they share the face v_2, and $e_1 \sim e_2$ because they share the face v_1. Further $e_1 \simeq e_2$ because their hull $t_1 \in C$, but $e_1 \not\simeq e_4$, so that $e_1 \underset{1}{\sim} e_4$. We have $t_1 \sim t_2$ because they share the face e_3, and also $t_1 \underset{2}{\sim} t_2$.

When $\kappa > \kappa_{\max}$, $C_\kappa = \emptyset$. We also refer to C as a *κ-complex* to note that $\kappa_{\max}(C) = \kappa$. Let a *$\kappa$-region* of C refer to any non-empty subset of C_κ, and let a κ-subregion refer to any subset of a κ-region.

Let C be a simplicial complex, and $\sigma, \tau \in C_\kappa$, for some $\kappa > 0$. When σ, τ share a face, they are *weakly adjacent*, denoted by $\sigma \sim \tau$. Their shared boundary face is denoted $\mathrm{bd}(\sigma, \tau)$. If $\sigma \sim \tau$, and in addition, they both share a co-face, this co-face is called their *hull*, denoted by $\mathrm{hl}(\sigma, \tau)$. If $\sigma, \tau \in C$, $\sigma \sim \tau$, and $\mathrm{hl}(\sigma, \tau) \in C$, then σ, τ are *strongly adjacent*, denoted by $\sigma \simeq \tau$. If $\sigma \sim \tau$, but $\sigma \not\simeq \tau$ in C, then σ, τ are *κ-adjacent*, denoted $\sigma \underset{\kappa}{\sim} \tau$.

n *oriented simplex* σ further has an orientation $p_\sigma \in \{\pm 1\}$, which indicates whether its parity with its faces is the same as, or opposite to, its natural parity. When $p_\sigma = +1$, we say σ is in *natural orientation*. For example, a directed edge $e = (v_i, v_j)$ for $i < j$ is in natural orientation, while if $i > j$, $p_e =$ 1. n *oriented simplicial complex* is a simplicial complex whose members are oriented. n oriented complex C contains at most one orientation for any given simplex, in the sense that $\sigma, \sigma' \in C, V(\sigma) = V(\sigma') \implies p_\sigma = p_{\sigma'}$.

Natural orientation does not account for simplex orientation. "Flipping" the directed edge e corresponds to switching the sign of p_e on the one hand, and to switching the head and tail vertices of e on the other. nalogously, the *head set* σ^{h} and *tail set* σ^{t} of an oriented simplex σ consist of those faces α satisfying p $\mathrm{nat}(\sigma, \alpha) > 0$ and p $\mathrm{nat}(\sigma, \alpha) < 0$, respectively. These sets are also balanced, as with the natural heads/tails. Finally, by introducing face orientations as well, we consider the *relative orientation* of an oriented simplex σ with respect to its oriented face α, as

simply

$$\mathrm{rel}(\sigma, \alpha) \coloneqq p_\sigma p \quad \mathrm{nat}(\sigma, \alpha) \ .$$

Let $G = (V, E)$ be a simple directed graph. We may always associate an oriented simplicial 1-complex $\tilde{G}$ with G, as follows. Enumerate $V = \{v_1, \ldots, v_n\}$. For each $v_i \in V$, we have $(\{v_i\}, +1) \in \tilde{G}$, and for each $(v_i, v_j) \in E$, we have $(\{v_i, v_j\}, \mathrm{sgn}(j \quad i)) \in \tilde{G}$. With this convention, the orientation of the 1-simplex corresponds to the agreement of the edge direction with the global ordering of the vertices.

Further, there are many possible ways to construct higher-order complexes that represent higher-order structure of the graph G. For example, we may construct the *κ-clique complex* $K_\kappa(G)$ for each $\kappa \in \mathbb{N}$, where $\kappa_{\max}(K_\kappa(G)) \leq \kappa$, as follows. For each fully-connected subgraph of G on vertices $v_{\ell_1}, \ldots, v_{\ell_{\kappa'}}$ for $\kappa' \leq \kappa + 1$, we have

$$\left(\{v_{\ell_1}, \ldots, v_{\ell_{\kappa'}}\}, \prod_{1 \leq i < j \leq \kappa'} \mathrm{sgn}(\ell_j \quad \ell_i) \right) \in K_\kappa(G) \ .$$

Notice that $\tilde{G} = K_1(G)$; additionally, we will abuse notation and refer to both G and $\tilde{G}$ as G when the usage is clear from context. While there are many simplicial complexes that represent a graph which are not κ-clique complexes, because simplicial complexes are closed under taking subsets of the underlying unoriented simplices, whenever $\delta \in C$, the complete subgraph on the vertices in δ is a subgraph of G.

Let X_κ be the space of real-valued functions on C_κ for each $\kappa \in \{0, 1, \ldots, \kappa_{\max}(C)\}$. If we wished to lift the non-degeneracy condition on C, we could instead restrict each X_κ to the set of *alternating* functions on C_κ, such that for $f \in X_\kappa$ and $\sigma, \tau \in C$, we have $V(\sigma) = V(\tau) \implies p_\sigma f(\sigma) = p_\tau f(\tau)$. In the case of graphs, X_0 consists of functions taking values on vertices, or graph signals. X_1 consists of functions on edges, or edge flows. function in X_1 is positive when the corresponding flow direction agrees with the edge orientation, and negative when the flow disagrees. n alternating function $f \in X_1$ would satisfy $f(\tilde{e}) = \quad f(e)$, for e a directed edge and $\tilde{e}$ the flipped edge. X_2 consists of functions on oriented triangles.

Given an oriented simplicial complex C, for each $\kappa \in \{0, 1, \ldots, \kappa_{\max}\}$, the *boundary operator* is a linear operator $B_\kappa : X_{\kappa+1} \mapsto X_\kappa$, where for $\sigma \in C_{\kappa+1}$, $\alpha \in C_\kappa$, the corresponding matrix

entries are $[B_\kappa]_{\ \sigma} = \mathrm{rel}(\sigma, \alpha)$. Likewise, the *coboundary operator* for each $\kappa \in \{0, 1, \ldots, \kappa_{\max}\}$ is just $B_\kappa : X_\kappa \to X_{\kappa+1}$, the adjoint to B_κ. The expression of the entries of B_κ as the relative orientation between simplex and face suggests that these are a natural way to construct functions taking local signed averages, according to adjacency in the simplicial complex.

In discrete differential geometry, B_κ corresponds to the *discrete exterior derivative*, is interpreted as the *discrete differential* **[65]**, and acts as a first-order difference operator of the graph signal. Particularly, for a graph signal $\boldsymbol{f}$, and a directed edge $e = (u, v)$ with $e^{\mathrm{h}} = \{v\}$, $e^{\mathrm{t}} = \{u\}$,

$$[B_0\, \boldsymbol{f}]_e = p_v[\boldsymbol{f}]_v \quad p_u[\boldsymbol{f}]_u \ . \tag{2.21}$$

In the standard graph setting, $p_u = p_v = 1$, and so $[B_0\, \boldsymbol{f}]_e = [\boldsymbol{f}]_v \quad [\boldsymbol{f}]_u$, and the resulting edge-valued function acts as the discrete gradient. Likewise, B_κ can be interpreted as the *discrete codifferential*. nalogously, for an edge flow $\boldsymbol{F}$, and a vertex v,

$$[B_0 \boldsymbol{F}]_v = p_v \left(\sum_{e: e^{\mathrm{h}} = \{v\}} \quad \sum_{e: e^{\mathrm{t}} = \{v\}} \right) [\boldsymbol{F}]_e \ , \tag{2.22}$$

and taking $p_v = 1$, $[B_0 \boldsymbol{F}]_v = \left(\sum_{e: e^{\mathrm{h}} = \{v\}} \quad \sum_{e: e^{\mathrm{t}} = \{v\}} \right) [\boldsymbol{F}]_e$. The resulting graph signal acts as the discrete divergence of the edge flow.

For the next calculation only, given a directed edge e, let u_+ refer to its head, and u refer to its tail. Combining Eqs. (2.21, 2.22),

$$\begin{aligned}
[B_0 B_0\, \boldsymbol{f}]_v &= p_v \left(\sum_{e: e^{\mathrm{h}} = \{v\}} \quad \sum_{e: e^{\mathrm{t}} = \{v\}} \right) [B_0\, \boldsymbol{f}]_e \\
&= p_v \left(\sum_{e: e^{\mathrm{h}} = \{v\}} \quad \sum_{e: e^{\mathrm{t}} = \{v\}} \right) \ p_{u_+}[\boldsymbol{f}]_{u_+} \quad p_u \ [\boldsymbol{f}]_u \ ) \\
&= p_v \left(\sum_{e: e^{\mathrm{h}} = \{v\} \text{ or } e^{\mathrm{t}} = \{v\}} p_v[\boldsymbol{f}]_v \quad \sum_{u: e = (u,v) \text{ or } e = (v,u)} p_u[\boldsymbol{f}]_u \right) \\
&= [\boldsymbol{d}]_v[\boldsymbol{f}]_v \quad \sum_{u \sim v} p_u p_v [\boldsymbol{f}]_u \ .
\end{aligned} \tag{2.23}$$

In Eq. (2.23), we see that each sum simply yields a sum over vertices adjacent to v, and that the first just counts them, yielding the usual degree vector, whose entries we denote $[\boldsymbol{d}]_v$. Taking $p_u = 1$ for all vertices, we retrieve the usual graph Laplacian, and recover both the standard incidence matrix identity $B_0 B_0 = L_0$, and the standard differential geometry identity div grad = . Conversely, in general, we foreshadow a phenomenon underlying much of the simplicial setting: while edge orientation in the incidence matrix does not affect the resulting Laplacian, this is only the case for *natural orientations* of edges with respect to vertices, and further re-orienting vertices may yield a different Laplacian.

Signal processing on simplicial complexes arises as a natural problem in the setting where richer structure is incorporated in data, than just scalar functions and pairwise relationships [**83**]. Given a discrete domain, data may include higher-order interactions between vertices, and so naturally be represented as functions or signals on a hypergraph [**9**]. In [**50**], matrix data is represented as signals on a bipartite graph indexed by the matrix's dimension labels. Similarly, κ-tensor data can be represented as signals on a κ-uniform, κ-partite hypergraph indexed by the tensor's dimension labels. One may always associate a κ-simplex with a hyperedge of cardinality $\kappa + 1$ in a hypergraph; when orientations are additionally part of the data or have a natural interpretation, and when data can be naturally interpreted on sub- or supersets of the vertices in a hyperedge, then it becomes useful to frame the relevant data processing as a problem of signal processing on an oriented simplicial complex.

Higher-order analysis is also possible with spatial data, scattered data, and point clouds. simplicial complex may be efficiently constructed on a metric space, for example, as in the Vietoris-Rips complex [**102**], where for a fixed distance d, a simplex σ is present in the complex whenever the diameter of σ does not exceed d. Efficient algorithms exist for constructing various simplicial complexes from graph data, such as clique, Vietoris-Rips, and witness complexes, and for performing common operations on simplicial complexes, such as the *simplex tree* data structure [**10**], a simplified version of which we use to represent simplicial complexes. The tree represents every simplex in the complex as one vertex, and is rooted, with the root representing the empty set. Each facet (i.e., maximal simplex) is represented by a leaf of the tree, and each vertex is labeled with the maximum vertex index not yet labeled by any of its ancestor vertices. In Figure 2.7, the tree structure and

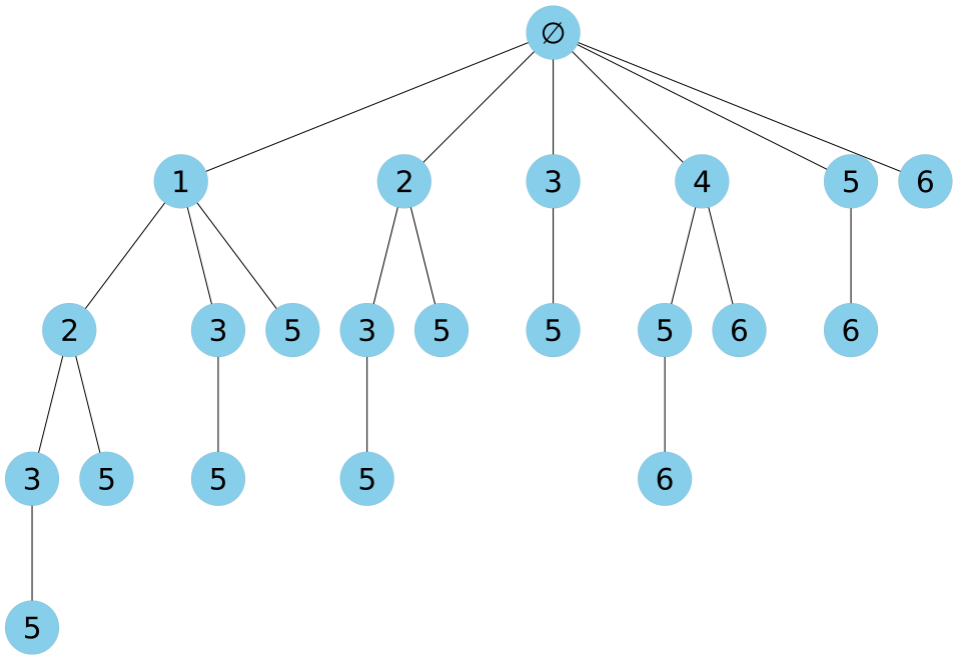


FIGURE 2.7. The simplex tree representation for a simplicial complex whose maximal simplices have labels $\{1,2,3,5\}$ and $\{4,5,6\}$. Each vertex represents a simplex in the complex, and if the length of the path to the root labeled $\emptyset$ is ℓ, then it is an ℓ-simplex. The labels for the vertices in the simplex are obtained by accumulating the labels on the path to that simplex from the root. So for example, the 3-simplices in this complex are $\{1,2,3\}, \{1,2,5\}, \{1,3,5\}, \{2,3,5\}$, and $\{4,5,6\}$

labels are demonstrated for a simplicial complex whose maximal simplices have labels $\{1,2,3,5\}$ and $\{4,5,6\}$. Because for a $(\kappa+1)$-simplex σ with face α, $[B_\kappa]_{\sigma} = p_\sigma p \ \mathrm{nat}(\sigma,\alpha)$, for a given complex the elements of this sparse matrix can be efficiently computed by traversing the complex's simplex tree while observing the orientation of each simplex encountered.

While in the $\kappa = 0$ case, the spectral analysis pipeline generally involves nothing more complicated than performing linear algebra on easy-to-assemble adjacency matrices, naive operations with κ-regions and adjacencies risk blowing up in cost, polynomially as a function of κ, both in memory and time, and so require more care. Once a simplicial complex is either given or computed as above, in simplex tree form, the next step is to represent the κ-adjacency data, degrees, and possibly weights, in a sparse fashion, i.e., iterating over $\mathcal{O}(|C_{\kappa+1}|)$ hulls instead of $O\left(|C_\kappa|^2\right)$ simplex pairs, most of which are likely not adjacent. The data structure `KRegion` we use to represent these

adjacencies is, for several reasons we will go into in Chapter 4, the most complicated we needed to develop.

CH PTER 3

Discrete Integral Operators

In this chapter, before diving into the dissertation's primary focus on functions living on κ-regions of simplicial complexes, we first develop a theory of spectral partitioning for functions on graphs via discrete integral operators as an appealing alternative to the usual Laplacian-based spectral theory. This work is part of an ongoing, yet unfinished project, to unify the construction of multiscale transforms on κ-regions, with the capabilities of discrete integral operator-based spectral embeddings and partitioning. That project spans exterior algebra, discrete and continuous differential geometry, discrete and continuous Hodge theory, harmonic analysis, potential theory, and spectral geometry, which we turn towards the usual problems of discrete signal processing. In this dissertation, we establish the first concrete step, by clearly justifying, constructing, and demonstrating discrete integral operators suitable for performing spectral embedding and graph partitioning.

3.1. Overview

Fourier analysis entails using an orthonormal basis for representation of data or functions on domains with some regularity, in the sense that they may be periodically extended; tensor products of intervals and lattices are typical domains. The perspective from harmonic analysis is to generalize this representation by relaxing the conditions we place on the domain, usually by using the eigenfunctions of an appropriate linear operator as the basis. For instance, for an open, connected set $\Omega \subset \mathbb{R}^d$ with a sufficiently smooth boundary $\partial\Omega$, we may take the Laplace operator (or Laplacian) $:= \sum_i \partial_{x_i}^2$ acting on $C^2(\Omega)$ functions satisfying self-adjoint boundary conditions on $\partial\Omega$. lternatively, we can consider the corresponding integral operator against a Green's function of this Laplacian. Because these operators invert each other, they commute, and so we obtain the same eigenbasis.

In the case of graphs more general than lattices, a canonical choice of operator is the graph Laplacian matrix $L := D \quad W$, or its normalized form $L^{\text{sym}} := D \ ^{\frac{1}{2}} L D \ ^{\frac{1}{2}}$, where W is the (possibly weighted) adjacency matrix of the graph, and $D = \text{diag}(W\mathbf{1})$. s reviewed in Section 2.2, a fundamental principle of spectral graph theory is that the eigenvalues of L and L^{sym} yield information on the structure of the graph. For example, the second-smallest eigenvalue of L^{sym} is a measure of the connectivity of a graph, and if we partition the vertices according to the sign of the corresponding eigenvector, we obtain approximately equally-sized subsets with few edges between them, ideal for image segmentation or clustering applications **[16, 87, 96]**.

Using the continuous Laplacian in computational applications requires performing some discretization, followed by application of the ideas from the graph case. For example, the matrix formed by applying finite difference approximation to the continuous Laplacian is precisely the graph Laplacian, when Neumann boundary conditions are imposed. In general however, numerical differentiation is well-known to be ill-conditioned, so another option is to discretize an integral operator via a quadrature formula. In one dimension (without loss of generality, on $(0, 1)$), given quadrature node and weight vectors $\boldsymbol{x}$ and $\boldsymbol{w}$ respectively, and an integral operator $\mathcal{G}$ with kernel G, we can approximate[1] via

$$\mathcal{G}f([\boldsymbol{x}]_i) = \int_0^1 G([\boldsymbol{x}]_i, y) f(y) \, \mathrm{d}y \approx \sum_j G([\boldsymbol{x}]_i, [\boldsymbol{x}]_j) f([\boldsymbol{x}]_j) [\boldsymbol{w}]_j \ .$$

dopting notation we will repeatedly use, a scalar function of one variable given a vector argument $\boldsymbol{x}$, like $f(\boldsymbol{x})$, will refer to the vector of element-wise evaluations. Similarly, for vector $\boldsymbol{x}$ and scalar y, the expression $G(\boldsymbol{x}, y)$ refers to the vector $[G(\boldsymbol{x}, y)]_i = G([\boldsymbol{x}]_i, y)$, and $G(\boldsymbol{x}, \boldsymbol{x})$ refers to the matrix $[G(\boldsymbol{x}, \boldsymbol{x})]_{ij} = G([\boldsymbol{x}]_i, [\boldsymbol{x}]_j)$. So, we may write

$$\mathcal{G}f(\boldsymbol{x}) = \int_0^1 G(\boldsymbol{x}, y) f(y) \, \mathrm{d}y \approx G(\boldsymbol{x}, \boldsymbol{x}) \, \text{diag}(\boldsymbol{w}) f(\boldsymbol{x}) \ .$$

In particular, for each quadrature formula on k nodes, we can construct a linear operator (here, the matrix $G(\boldsymbol{x}, \boldsymbol{x}) \, \text{diag}(\boldsymbol{w})$) on $\mathbb{R}^k$ which represents the integral operator $\mathcal{G}$ on $(0, 1)$; these are examples of discrete integral operators. For example, consider the kernel $G(x, y) = |x \quad y|$, for $x, y \in (0, 1)$. For a quadrature formula with uniform weights, the entries of the corresponding

[1] The meaning of ‘ ‘ here depends on the accuracy of the quadrature formula used.

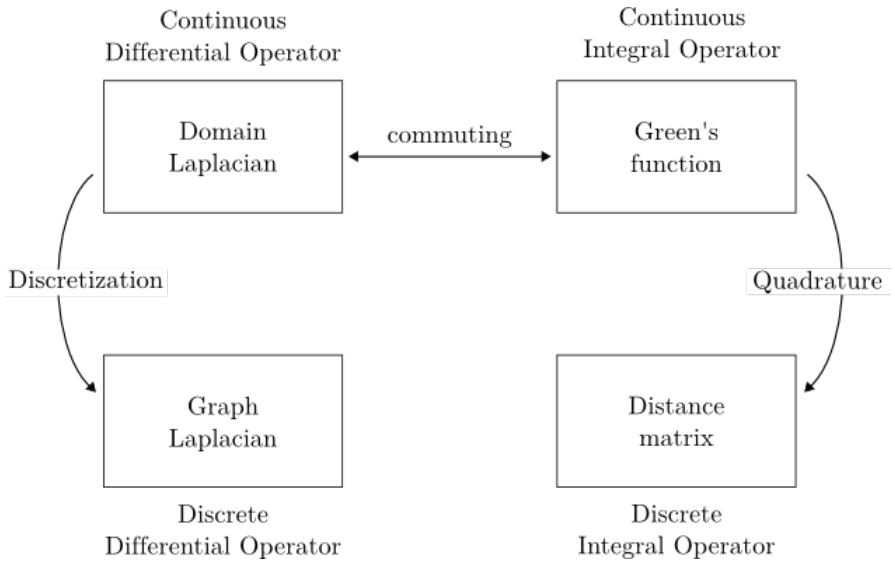


FIGURE 3.1. Diagram showing the relationship between the various kinds of operators whose eigenfunctions may be used for analysis of the underlying domains, with particular instances.

discrete integral operator are each Euclidean distances between the quadrature nodes, and so we obtain the shortest-path distance matrix of the graph created (in this case, a path graph).

The distance matrix demonstrates typical properties of discrete integral operators – they contain *global* information about a graph, as opposed to the *local* adjacency information in a graph Laplacian. In the case of directed graphs, this discrepancy is exaggerated, as adjacency is no longer a local property of the graph, and is best represented by the existence of a shortest directed path.

To concretely illustrate the main ideas at play, we will demonstrate all four cases of the operators mentioned, along with their respective orthonormal eigenbases. Figure 3.1 summarizes the relationships among operators considered.

3.2. Potential Theory

3.2.1. One-dimensional continuous and path-graph Laplacians.

The classic Fourier basis may be expressed as the family of functions $e_n(x) := \exp(2\pi i n x)$ on $[0,1] \subset \mathbb{R}$, for $n \in \mathbb{Z}$. It is well known that $\{e_n\}_{n\in\mathbb{Z}}$ forms a complete orthonormal basis for $L^2([0,1])$; these are the Laplacian eigenfunctions on $[0,1]$ satisfying the periodic boundary conditions $f(0) = f(1)$, $f'(0) = f'(1)$. Likewise, enforcing the Dirichlet boundary conditions $f(0) = f(1) = 0$

yields the basis $s_n(x) := \sqrt{2}\sin n\pi x$ for $n \in \mathbb{N}$, and enforcing the Neumann boundary conditions $f'(0) = f'(1) = 0$ yields the basis $c_n(x) := \sqrt{2}\cos n\pi x$ for $n \in \mathbb{N}$ and $c_0(x) \equiv 1$.

In each case, via integration by parts we can find an integral operator which inverts $\mathcal{L}_1 :=$, the one-dimensional (negative) Laplacian, on the class of functions on $[0,1]$ satisfying the boundary conditions:

$$\text{(3.1)} \qquad \begin{cases} \text{Dirichlet:} & G_D(x,y) = \min\{x,y\} \quad xy = \quad \frac{1}{2}|x \quad y| + \frac{x+y}{2} \quad xy \\ \text{Neumann:} & G_N(x,y) = \quad \frac{1}{2}|x \quad y| + \frac{1}{2}(x \quad \frac{1}{2})^2 + \frac{1}{2}(y \quad \frac{1}{2})^2 + \frac{1}{12} \\ \text{Periodic:} & G_{\text{per}}(x,y) = \quad \frac{1}{2}|x \quad y| + \frac{(x \quad y)^2}{2} + \frac{1}{12}\ . \end{cases}$$

Now, let $n \in \mathbb{N}$. We can use the n midpoint nodes $[\boldsymbol{x}]_k = \frac{2k \quad 1}{2n}$ for $k = 1 : n$, to construct a path graph P_n which will represent the discrete version of the interval. Recall that $\mathbf{1}$ is the constant vector, with $[\mathbf{1}]_i = 1$. The path-graph Laplacian has the usual form

$$L = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 2 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 2 & 1 \\ 0 & \cdots & 0 & 0 & 1 & 1 \end{bmatrix},$$

which matches the finite difference approximation of $\mathcal{L}_1$, assuming Neumann boundary conditions. The Laplacian eigenvectors of P_n are precisely the discrete cosine transform (DCT-II) vectors, taking the form $\boldsymbol{v}_k \propto \cos k\pi\boldsymbol{x}$ for $k = 0 : n \quad 1$ [**91**]. These eigenvectors $\boldsymbol{v}_k$, when scaled appropriately, interpolate the eigenfunctions c_k of the continuous Neumann Laplacian at $\boldsymbol{x}$, for $k = 0 : n \quad 1$. If we assume different boundary conditions, only the first and last rows and columns of the resulting matrix change. In the Dirichlet case, the eigenvectors are precisely the discrete sine transform (DST-II) vectors, and interpolate the eigenfunctions s_k at $\boldsymbol{x}$, for $k = 1 : n$. In the periodic case, the eigenvectors are the discrete Fourier transform (DFT) vectors, and can be scaled to interpolate the eigenfunctions e_k at either the midpoints ($\boldsymbol{x}$) or the gridpoints ($\boldsymbol{x} \quad \frac{1}{2n}\mathbf{1}$). In summary, in

this special one-dimensional case, the vectors formed by evaluating orthogonal eigenfunctions at midpoint nodes are themselves (discretely) orthogonal.

lternatively, we can use the midpoint quadrature, with nodes $\boldsymbol{x}$ (the midpoints as above), and weights $\boldsymbol{w} = \frac{1}{n}\mathbf{1}$, to discretize the Green's functions in (3.1). We would like to determine the relationship between the path-graph Laplacian above, and the discrete integral operator corresponding to the Neumann Green's function. Given an integral operator $\mathcal{G}$ with kernel G, the eigen-pairs of the matrix $G(\boldsymbol{x}, \boldsymbol{x})$ can be used to approximate those of $\mathcal{G}$, in the sense that

$$\mathcal{G}f = \lambda f \quad \text{and} \quad \mathcal{G}f(\boldsymbol{x}) \approx \frac{1}{n} G(\boldsymbol{x}, \boldsymbol{x}) f(\boldsymbol{x}) \quad \implies \quad G(\boldsymbol{x}, \boldsymbol{x}) f(\boldsymbol{x}) \approx n\lambda f(\boldsymbol{x}) \ .$$

However, by itself this offers no guarantee that $f(\boldsymbol{x})$ is an approximate eigenvector of $G(\boldsymbol{x}, \boldsymbol{x})$, a significant drawback. Further, for the term in common to each Green's function in (3.1), highlighted in red, we can identify its matrix of discrete values as a distance matrix for the path graph P_n, but the other terms are more difficult to analyze. This term is the *fundamental solution* of the Laplacian; we can use this observation to overcome both difficulties, by analyzing the corresponding integral operator instead. To see how, we will move to the general case.

3.2.2. Boundary conditions for the integral operator with harmonic kernel.

Let us consider the formal differential operator $\mathcal{L} :=$ operating over a sufficiently smooth bounded domain $\Omega \subset \mathbb{R}^d$ for some $d \geq 1$. Let $B_\epsilon(x)$ refer to the open ball of radius ϵ centered at x in $\mathbb{R}^d$. Recall the Cauchy principal value p.v., defined for an integral operator $\mathcal{G}$ with kernel G, where $G(x, y)$ is singular when $x = y$, by

$$\text{p.v.}\, \mathcal{G}f(x) \coloneqq \lim_{\epsilon \to 0} \int_{\mathbb{R}^d \setminus B_\epsilon(x)} G(x, y) f(y) \, \mathrm{d}y \ .$$

Let ∂_ν refer to the outward normal derivative on $\partial\Omega$, and let $y \mapsto \sigma(y)$ be the surface measure on $\partial\Omega$. relation sometimes called *Poisson's integral formula* [**54**, Ch. 4 Eq. 3.7] states that for $f \in C^1(\overline{\Omega})$, we have $\mathcal{L}\mathcal{K}f = f$, where $\mathcal{K}$ is the integral operator

$$\mathcal{K}f(x) := \int_\Omega K(x, y) f(y) \, \mathrm{d}y \ ,$$

against the *harmonic kernel*

$$K(x,y) = \begin{cases} \frac{1}{2}|x \quad y| & d = 1 \\ \frac{1}{2\pi} \ln |\boldsymbol{x} \quad \boldsymbol{y}| & d = 2 \\ \frac{1}{(2 \quad d)\omega_d} |\boldsymbol{x} \quad \boldsymbol{y}|^{2 \quad d} & d \geq 3 \ . \end{cases}$$

Here $\omega_d := \frac{2\pi^{d/2}}{(d/2)}$ is the surface area of the unit ball in $\mathbb{R}^d$ and $|\cdot|$ is the Euclidean norm. Further, for $f \in C^2(\Omega) \cap C^1(\overline{\Omega})$, we can use Green's identities [**54**, Ch. 4 Eq. (1.2)] to find that

$$\text{p.v.}\, \mathcal{K}\mathcal{L}f(x) = \text{p.v.} \int_\Omega K(x,y)\mathcal{L}f(y)\,\mathrm{d}y = f(x) \quad \int_{\partial\Omega} \left[K(x,y)\partial_{\nu(y)}f(y) \quad f(y)\partial_{\nu(y)}K(x,y)\right]\,\mathrm{d}\sigma(y) \ .$$

Because K is weakly singular, $\mathcal{K}$ is compact, and so the function on the LHS exists and the principal value is unnecessary [**61**, Theorem 2.29]. In [**74**], it is observed that the functions f for which $\mathcal{K}\mathcal{L}f = \mathcal{L}\mathcal{K}f = f$, satisfy the *non-local boundary conditions* (NLBC):

$$\int_{\partial\Omega} K(x,y)\partial_{\nu(y)}f(y)\,\mathrm{d}\sigma(y) = \quad \frac{1}{2}f(x) + \text{p.v.} \int_{\partial\Omega} f(y)\partial_{\nu(y)}K(x,y)\,\mathrm{d}\sigma(y) \quad \text{for each} \quad x \in \partial\Omega \ . \tag{3.2}$$

How restrictive of a set of boundary conditions are the NLBC? Is $\mathcal{L}$ equipped with the NLBCs essentially self-adjoint? The answer is known to be affirmative for $d = 1, 2$, and we suspect it to be true in general, which we formalize in a conjecture; we will assume this conjecture is true for the remainder of the chapter.

ONJECTURE 3.2.1. *Let $\mathcal{S} := C^2(\Omega) \cap C^1(\overline{\Omega})$ consist of sufficiently smooth functions on $\Omega \subset \mathbb{R}^d$, a sufficiently smooth domain; say, $\partial\Omega \in C^2$, using the notation of [**6**]. Let $\mathcal{L}_1 := (\mathcal{D}, \mathcal{L})$ denote the unbounded differential operator acting on the domain $\mathcal{D} := \{f \in \mathcal{S} \mid f$ satisfies NL $C\}$. Then $\mathcal{L}_1$ is essentially self-adjoint, has a complete set of eigenfunctions with nonzero eigenvalue, and has $\mathcal{K}$ as its inverse.*

Remarks. The critical assumptions which Conjecture 3.2.1 rely on are sufficient smoothness of the eigenfunctions of $\mathcal{K}$, and the empty null space of $\mathcal{L}_1$, and so of $\mathcal{K}$. In the cases $d = 1, 2$, it is known that $\mathcal{K}$ has a trivial null space [**40**, **94**]. For $d \geq 3$, a literature search did not return a clear positive result, but [**61**, Theorem 4.20] is suggestive. Since $\mathcal{K}$ is compact and self-adjoint, we may find an orthonormal basis of $L^2(\Omega)$ consisting of its eigenfunctions, and the only potential

non-eigenvalue in the spectrum is $\lambda = 0$; hence the importance of the null space. Kress, in turn citing Hähner, explains a framework where for integral operators with a weakly singular kernel, of which $\mathcal{K}$ is a principal example, the Fredholm alternative can be used to lift the eigenfunctions from just L^2 to C^0; i.e., continuity. [**61**, Theorems 7.5, 7.6] also develop Hölder continuity $C^{1,}$ of the single- and double-layer potentials defined on $\partial\Omega$ in Eq. (3.2). The smoothness class $\mathcal{S}$ is chosen such that the Poisson and Green's identities hold, such that $f \in \mathcal{S} \implies \mathcal{L}\mathcal{K}f = f$, and the punchline is that if the eigenfunctions of $\mathcal{K}$ are known to be in S, then when (λ, f) is an eigenpair of $\mathcal{K}$, $\mathcal{K}f = 0 \implies \mathcal{L}\mathcal{K}f = f = 0$, and so $\lambda \neq 0$. If we assume the lifting strategy can be extended to $\mathcal{S}$, or otherwise that the eigenfunctions of $\mathcal{K}$ are in S, and so the null space of $\mathcal{K}$ is trivial for $d \geq 3$, then most of the conjecture follows through a straightforward application of the spectral theorem. We stop short of proving essential self-adjointness, though it seems quite likely.

PROPOSITION 3.2.1. *Let $\mathcal{S}, \mathcal{L}_1$ be as in Conjecture 3.2.1, and suppose that the eigenfunctions of $\mathcal{K}$ are in $\mathcal{S}$. Then $\mathcal{L}_1$ is symmetric, has a complete set of eigenfunctions with nonzero eigenvalue, and has $\mathcal{K}$ as its inverse.*

PROOF. The notation here is the same as in Conjecture 3.2.1. s $\mathcal{K}$ is compact and self-adjoint, the eigenvalues of $\mathcal{K}$ have 0 as a limit point, and 0 is not in the spectrum, as $\mathcal{K}f = 0 \implies \mathcal{L}\mathcal{K}f = f = 0$. Hence $\mathcal{K}$ has a complete set of eigenpairs $\{(f_i, \lambda_i)\}_i$, and so $\mathcal{L}$ has a complete set of eigenpairs $\{(f_i, 1/\lambda_i)\}_i$. Because $\mathcal{L}$ and $\mathcal{K}$ commute for each f_i, they satisfy the NLBC (3.2), and so the f_i form an orthonormal basis for $\mathcal{D}$. Hence, the formal operator $\mathcal{L}_1$ is symmetric, as $f, g \in \mathcal{D} \implies \langle \mathcal{L}f, g\rangle = \langle f, \mathcal{L}g\rangle$, with $\mathcal{L}g \in C^0(\Omega)$. $\diamond$

Regarding essential self-adjointness, we know that $C^0(\Omega)$ is dense in $L^2(\Omega)$, and that the precise domain of g that satisfy $\mathcal{L}g \in L^2(\Omega)$ is the *Sobolev space* $H^2(\Omega)$ [**1**]. In principle, passing from functions to distributions, and from ordinary to weak derivatives, an analogous argument to the proof of Proposition 3.2.1 goes through, from which we could conclude that $\mathcal{L}_1$ is essentially self-adjoint. We stop short of this claim because we do not know whether $g \in H^2(\Omega)$ guarantees the rest of the assumptions, that g satisfies the NLBC.

The NLBC has the desirable property that the functional form of $\mathcal{K}$ is independent of the domain Ω. In order to find an integral operator which commutes with functions satisfying the Neumann

condition $\partial_\nu f \equiv 0$ on $\partial\Omega$, naturally we cannot hope for such a happy coincidence. One consequence of this is that no 'interesting' functions satisfy both the Neumann conditions and the NLBC, in the sense that all such functions satisfy both the Neumann and Dirichlet conditions.

PROPOSITION 3.2.2. *Consider the following three subspaces of $C^2(\Omega) \cap C^1(\overline{\Omega})$ functions:*

$$N := \{f \,|\, \partial_\nu f \equiv 0 \text{ on } \partial\Omega\} \quad , \quad B := \{f \,|\, f \text{ satisfies NL } \ C\} \quad , \text{ and } \quad D := \{f \,|\, f \equiv 0 \text{ on } \partial\Omega\} \ .$$

Then we have that $B \cap N \subset D$ and $B \cap D \subset N$.

PROOF. First, let us define operators $R, T : C^1(\partial\Omega) \mapsto C^1(\partial\Omega)$, such that

$$Rf(x) = \int_{\partial\Omega} f(y) K(x,y) \, \mathrm{d}\sigma(y) \quad \text{and} \quad Tf(x) = \text{p.v.} \int_{\partial\Omega} f(y) \partial_{\nu(y)} K(x,y) \, \mathrm{d}\sigma(y) \ .$$

These are similar to single/double layer potentials, but defined on the boundary. Note the sign difference between corresponding operators in **[32]**. Now, suppose $f \in B \cap N$. Then, we must have that

$$0 = \quad \frac{1}{2} f(x) + \text{p.v.} \int_{\partial\Omega} f(y) \partial_{\nu(y)} K(x,y) \, \mathrm{d}\sigma(y) = \quad \left(\frac{1}{2} I \quad T \right) f(x) \quad \text{for } x \in \partial\Omega.$$

It is well-known that the double-layer potential on Ω generated by a boundary function g extends continuously to the boundary with the value $(\frac{1}{2}I \quad T)g$ **[32**, Theorem 3.22]. This potential solves the Dirichlet boundary value problem, of finding a function harmonic on Ω whose boundary values coincide with a given function on $\partial\Omega$. Because this solution is unique, the map $(\frac{1}{2}I \quad T)$ is invertible **[32**, Theorem 3.40a]. Hence, we conclude that $f \equiv 0$ on $\partial\Omega$, so that $f \in D$.

Instead, suppose $f \in B \cap D$. Then analogously, we find that we require $R(\partial_\nu f) \equiv 0$ on $\partial\Omega$. It is well-known that the single-layer potential on Ω generated by a boundary function g extends continuously to the boundary with the value Rg **[32**, Theorem 3.25]. Hence $\partial_\nu f$ uniquely solves the Dirichlet problem for vanishing boundary data, and itself vanishes on $\partial\Omega$. Thus $f \in N$. $\diamond$

In light of this, we will compromise by seeking out an operator which commutes with *some* of the Neumann functions, and so obtain a weak dependence on the domain. Recall that for functions satisfying the Neumann boundary conditions, thanks to Green's identity, the range of $\mathcal{L}$ is orthogonal

to constant functions:

$$\langle \mathcal{L}f, \mathbf{1}_\Omega \rangle = \int_\Omega \mathcal{L}f = \int_{\partial\Omega} \partial_\nu f = 0 \ .$$

Now, let $\mathcal{P}$ be the orthogonal projection onto the orthogonal complement of the indicator function $\mathbf{1}_\Omega$; that is, $\mathcal{P}f(x) := f(x) \quad \frac{1}{|\Omega|}\int_\Omega f(y)\,\mathrm{d}y$. Then if f satisfies the Neumann boundary conditions, we have $\mathcal{P}\mathcal{L}f = \mathcal{L}f$.

Further, because the solution of the Neumann boundary value problem is unique up to a constant [**32**, Theorem 3.40c], the only harmonic functions that satisfy the Neumann boundary conditions are the constant functions. Hence, we should include the condition $\int_\Omega f = 0$, or $\mathcal{P}f = f$ in our definition of the Neumann class, with respect to finding a commuting integral operator. Essentially, we are taking a quotient by the 1D subspace of constant functions, since whenever $\mathcal{P}\mathcal{L}f = \mathcal{L}f$, f differs by a constant from a function g for which $\mathcal{P}\mathcal{L}g = \mathcal{L}g$ and $\mathcal{P}g = g$. Given an orthonormal basis for these functions, we may then later complete it by including $\mathbf{1}_\Omega$ as a basis vector.

The invariance of the Neumann functions under $\mathcal{P}$ described above suggests that the operator $\mathcal{G}_H := \mathcal{P}\mathcal{K}\mathcal{P}$ may commute with some of them.[2] It is instructive to see that $\mathcal{G}_H$ is also a self-adjoint integral operator, and to know its kernel. To wit, we compute

$$\begin{aligned}
\mathcal{P}\mathcal{K}f(x) = \mathcal{K}f(x) \quad \frac{1}{|\Omega|}\int_\Omega \mathcal{K}f(z)\,\mathrm{d}z &= \int_\Omega K(x,y)f(y)\,\mathrm{d}y \quad \frac{1}{|\Omega|}\int_\Omega\int_\Omega K(z,y)f(y)\,\mathrm{d}y\,\mathrm{d}z \\
&= \int_\Omega \left[K(x,y) \quad \frac{1}{|\Omega|}\int_\Omega K(z,y)\,\mathrm{d}z\right] f(y)\,\mathrm{d}y \\
\text{(Note the use of } K(z,y) = K(y,z)) \ &= \int_\Omega \left[K(x,y) \quad \frac{1}{|\Omega|}\mathcal{K}\mathbf{1}_\Omega(y)\right] f(y)\,\mathrm{d}y \ .
\end{aligned}$$

Let $F(y) := \frac{1}{|\Omega|}\mathcal{K}\mathbf{1}_\Omega(y)$, and $h := \frac{1}{|\Omega|}\int_\Omega F(y)\,\mathrm{d}y$. Because $\mathcal{P}$ and $\mathcal{K}$ are self-adjoint, we expect that $\mathcal{K}\mathcal{P}f(x) = \int_\Omega [K(x,y) \quad F(x)]f(y)\,\mathrm{d}y$; calculating explicitly, we find

$$\begin{aligned}
\mathcal{K}\mathcal{P}f(x) &= \int_\Omega K(x,y)\left(f(y) \quad \frac{1}{|\Omega|}\int_\Omega f(z)\,\mathrm{d}z\right)\,\mathrm{d}y \\
&= \int_\Omega K(x,y)f(y)\,\mathrm{d}y \quad \int_\Omega F(x)f(z)\,\mathrm{d}z \\
&= \int_\Omega \left[K(x,y) \quad F(x)\right] f(y)\,\mathrm{d}y \ .
\end{aligned}$$

[2]Here, H refers to the harmonic kernel.

as desired. Thus,

$$\begin{aligned}\mathcal{G}_H f(x) &= \int_\Omega [K(x,y) \quad F(x)] f(y)\,\mathrm{d}y \quad \frac{1}{|\Omega|}\int_\Omega\int_\Omega [K(z,y) \quad F(z)] f(y)\,\mathrm{d}y\,\mathrm{d}z \\ &= \int_\Omega [K(x,y) \quad F(x) \quad F(y)+h]\, f(y)\,\mathrm{d}y =: \int_\Omega G_H(x,y) f(y)\,\mathrm{d}y\ .\end{aligned}$$

In one dimension, it turns out that we recover the usual Neumann Green's function this way. For this special case, there is the important property that every connected open set is an open ball; hence without loss of generality we may take $\Omega = (a,b)$.

PROPOSITION 3.2.3. *Let $\Omega = (a,b)$. Then,*

$$G_H(x,y) = K(x,y) + \frac{1}{2(b \quad a)}\left(\left(x \quad \frac{a+b}{2}\right)^2 + \left(y \quad \frac{a+b}{2}\right)^2\right) + \frac{b \quad a}{12}\ .$$

PROOF. We can explicitly compute F and h. This yields

$$\begin{aligned}F(y) = \frac{1}{b \quad a}\int_a^b \frac{1}{2}|z \quad y|\,\mathrm{d}z &= \frac{1}{4(b \quad a)}\ (b \quad y)^2 + (y \quad a)^2) \\ &= \frac{1}{2(b \quad a)}\left(y \quad \frac{a+b}{2}\right)^2 \quad \frac{b \quad a}{8}\ ,\end{aligned}$$

and

$$h = \quad \frac{b \quad a}{8} \quad \frac{1}{2(b \quad a)^2}\int_a^b \left(y \quad \frac{a+b}{2}\right)^2 \mathrm{d}y = \quad \frac{b \quad a}{8} \quad \frac{2}{6(b \quad a)^2}\left(\frac{b \quad a}{2}\right)^3 = \quad \frac{b \quad a}{6}\ .$$

Because $2/8 \quad 1/6 = 1/12$, the constant term is correct and we have the result. $\diamond$

For $(0,1)$, this reduces precisely to G_N from (3.1). This means we may represent G_N entirely in terms of K, which is the key to obtaining the right discrete integral operator by quadrature.

We have just seen that we may write the one-dimensional Neumann Green's function as

$$G_N(x,y) = \int_0^1\int_0^1 K(x,y) \quad K(z,y) \quad K(x,w) + K(z,w))\,\mathrm{d}z\,\mathrm{d}w\ .$$

We can approximate G_N with another kernel, formed by using midpoint quadrature on the double integral:

$$\mathcal{M}G_N(x,y) := \frac{1}{n^2}\sum_{\ell=1}^{n}\sum_{m=1}^{n} \; K(x,y) \quad K([\boldsymbol{x}]_\ell, y) \quad K(x,[\boldsymbol{x}]_m) + K([\boldsymbol{x}]_\ell,[\boldsymbol{x}]_m)\big) \approx G_N(x,y) \; .$$

Let P denote the orthogonal projection onto the orthogonal complement of $\mathbf{1}$, given explicitly by $P := I \quad \frac{1}{n}\mathbf{1}\mathbf{1}^*$. We will abuse notation here and later, and use K to denote the matrix of evaluations of the harmonic kernel, so that $[K]_{ij} := K([\boldsymbol{x}]_i,[\boldsymbol{x}]_j) = \quad \frac{1}{2}|[\boldsymbol{x}]_i \quad [\boldsymbol{x}]_j|$. Then for each $i \in 1:n$, and f arbitrary, the integral operator against $\mathcal{M}G_N$ is

$$\begin{aligned}
&\int_0^1 \mathcal{M}G_N([\boldsymbol{x}]_i, y) f(y)\,\mathrm{d}y \\
&\qquad = \mathcal{G}_H f([\boldsymbol{x}]_i) \quad \frac{1}{n}\sum_\ell \mathcal{G}_H f([\boldsymbol{x}]_\ell) \quad \frac{1}{n}\left(\int_0^1 f(y)\,\mathrm{d}y\right)\left(\sum_m [K]_{im} \quad \frac{1}{n}\sum_\ell\sum_m [K]_{\ell m}\right) \\
&\qquad = [P(\mathcal{G}_H f(\boldsymbol{x}))]_i \quad \frac{1}{n}\left(\int_0^1 f(y)\,\mathrm{d}y\right)\sum_m [PK]_{im} \\
&\qquad = [P(\mathcal{G}_H f(\boldsymbol{x}))]_i \quad \left(\int_0^1 f(y)\,\mathrm{d}y\right)\frac{1}{n}[PK\mathbf{1}]_i \; .
\end{aligned}$$

pplying one last midpoint quadrature, and collecting the result into a vector, yields the approximation:

$$\begin{aligned}
\int_0^1 \mathcal{M}G_N(\boldsymbol{x}, y) f(y)\,\mathrm{d}y &\approx \mathcal{M}\left(P\mathcal{G}_H f(\boldsymbol{x}) \quad \left(\int_0^1 f(y)\,\mathrm{d}y\right)\frac{1}{n}PK\mathbf{1}\right) \\
&= P\mathcal{M}\int_0^1 \left(K(\boldsymbol{x}, y) \quad \frac{1}{n}K\mathbf{1}\right) f(y)\,\mathrm{d}y \\
&= \frac{1}{n}P\sum_\ell \left(K_{\circ\ell} \quad \frac{1}{n}K\mathbf{1}\right) f([\boldsymbol{x}]_\ell) \\
&= \frac{1}{n}P\left(Kf(\boldsymbol{x}) \quad \frac{1}{n}K\mathbf{1}\mathbf{1}^* \; f(\boldsymbol{x})\right) \\
&= \frac{1}{n}PKPf(\boldsymbol{x}) =: \frac{1}{n}Hf(\boldsymbol{x}) \; .
\end{aligned}$$

The punchline will be that the eigenvectors of H are precisely those of the path-graph Laplacian, which we demonstrate in Section 3.3, and so this calculation shows that $\mathcal{M}G_N$ is the correct kernel

from which to construct the Neumann discrete integral operator for the path graph. The proof of this fact depends crucially on K being proportional to the path graph distance matrix.

3.2.3. Projecting on $\mathcal{K}$ in higher dimensions.

In higher dimensions, it turns out that $\mathcal{G}_H$ does not commute with all of the Neumann eigenfunctions – instead, we obtain precisely those Neumann functions which are *constant* on the boundary. Herein, for a set of functions A and a formal unbounded operator T, let (A, T) denote T equipped with domain A.

PROPOSITION 3.2.4. *Let* $S = \{f \in C^2(\Omega) \cap C^1(\overline{\Omega}) \,\big|\, \partial_\nu f \equiv 0 \text{ on } S \text{ and } \mathcal{P}f = f\}$, *and let* $S_c = \{f \in S \,\big|\, f \equiv c \text{ on } \partial\Omega \text{ for some } c \in \mathbb{R}\}$. *Then* $\mathcal{G}_H$ *commutes with the Neumann Laplacian* $\mathcal{L}_N := (S, \mathcal{L})$ *for* $d = 1$, *while for* $d > 1$, *instead* $\mathcal{G}_H$ *commutes with the operator* $\mathcal{L}_c := (S_c, \mathcal{L})$.

PROOF. First, suppose $f \in S$, and let's consider $\mathcal{L}\mathcal{G}_H f$. We find that $\mathcal{L}\mathcal{G}_H f = \mathcal{L}\mathcal{P}\mathcal{K}\mathcal{P}f = \mathcal{L}\mathcal{K}f = f$. Note that in general, $\mathcal{L}\mathcal{P}g = \mathcal{L}g$, since $\mathcal{P}g$ differs from g by a constant. Now, let's consider $\mathcal{G}_H\mathcal{L}f$. We compute

$$\begin{aligned}\mathcal{G}_H\mathcal{L}f(x) = \mathcal{P}\mathcal{K}\mathcal{P}\mathcal{L}f(x) = \mathcal{P}\mathcal{K}\mathcal{L}f(x) &= \mathcal{P}\left[f(x) \quad \int_{\partial\Omega}\left[K(x,y)\partial_\nu f(y) \quad f(y)\partial_{\nu(y)}K(x,y)\right]\,\mathrm{d}\sigma(y)\right] \\ &= f(x) + \mathcal{P}\left[\int_{\partial\Omega} f(y)\partial_{\nu(y)}K(x,y)\,\mathrm{d}\sigma(y)\right]\,. \qquad (3.3)\end{aligned}$$

The single-layer potential vanished because $\partial_\nu f \equiv 0$ on $\partial\Omega$. Let us consider the double-layer potential $u(f;x) := \int_{\partial\Omega} \partial_{\nu(y)}K(x,y)f(y)\,\mathrm{d}\sigma(y)$. In order for the extraneous term $\mathcal{P}u(f;x)$ to vanish, $u(f;x)$ must be killed by the projection, and so must be a constant (possibly dependent on f).

In the case of $d = 1$, we may take $\Omega = (a, b)$ for some $a, b \in \mathbb{R}$. We have $K(x,y) = \frac{1}{2}|x \quad y|$, and so $\nabla_y K(x,y) = \frac{1}{2}\operatorname{sgn}(y \quad x)$. Then for any $x \in (a,b)$,

$$u(f;x) = \frac{1}{2}\left(\operatorname{sgn}(b \quad x)f(b) \quad \operatorname{sgn}(a \quad x)f(a)\right) = \frac{1}{2}(f(a) + f(b))\,,$$

a constant independent of x. This confirms the desired result for $d = 1$.

Now, suppose $d \geq 2$. s in the proof of Proposition 3.2.2, we know that $u(f;x)$ extends continuously to the function $(\frac{1}{2}I \quad T)f(x)$ on $\partial\Omega$. Because $T\mathbf{1}_{\partial\Omega} \equiv \frac{1}{2}$ [**32**, Proposition 3.19], if $f \equiv c$ on $\partial\Omega$, then $u \equiv c$ on $\overline{\Omega}$. Conversely, if $u \equiv c$ on Ω, then u is harmonic, and extends

continuously to the constant function c on $\overline{\Omega}$. In particular, because the solution to the Dirichlet problem is unique, $f \equiv c$ is the only solution. Thus the extraneous term in (3.3) vanishes precisely when f is constant on $\partial\Omega$, establishing that $\mathcal{G}_H$ commutes with $\mathcal{L}_c$ on S_c as desired. $\diamond$

Thus while for $d = 1$, we can obtain the continuous Neumann Laplacian and corresponding discrete graph Laplacian from the harmonic kernel by projection alone, without considering boundary conditions, this negative result guides us away from attempting this for more general domains. Instead, we will approach from the discrete side, via the graph distance matrix.

3.3. Distance Matrices

Returning to the concrete goal of constructing discrete integral operators, we begin by slightly restating a result of Bapat et al. from 2005. Given a matrix M, let M° refer to the modified Hadamard reciprocal with $[M^{\circ}]_{ij} = 1/[M]_{ij}$ when $M_{ij} \neq 0$, and 0 otherwise. Let $M^{\circ r}$ refer to the usual Hadamard power with $[M^{\circ r}]_{ij} = [M]_{ij}^r$.

THEOREM 3.3.1 (Theorem 2.1 (Bapat et al. [**5**])). *Let G be a weighted tree on n vertices $v_1, \ldots, v_n$, and $\widetilde{G}$ be the same tree with reciprocal weights, such that $W(\widetilde{G}) = W(G)^{\circ}$. Let B be the distance matrix of G, $K := \frac{1}{2}B$, and L be the graph Laplacian of $\widetilde{G}$. Define $[\ ]_i := 2\ [\mathbf{d}]_i$, with $[\mathbf{d}]_i$ the unweighted degree of v_i, $w_1, \ldots, w_{n-1}$ as the edge weights of G, and $\sigma_k := \sum_{j=1}^{k} \boldsymbol{w}_j$. Then*

$$K^{-1} = L - \frac{1}{\sigma_{n-1}} \mathbf{1}\mathbf{1}^{\top}.$$

Because G is a tree on n vertices, it has $n-1$ edges, and so $\sum_i [\mathbf{d}]_i = 2n-2$, meaning $\mathbf{1}^{\top}L\mathbf{1} = 2 \neq 0$. So, because $L\mathbf{1} = \mathbf{0}$, multiplying the theorem statement on the right by $\mathbf{1}$, we find that $K^{-1}\mathbf{1} \propto \mathbf{1}$, so that $K\mathbf{1} \propto \mathbf{1}$. If we instead multiply by K, thanks to the symmetry of K, we find that

$$I = LK + c\ \mathbf{1}\mathbf{1}^{\top} \quad \text{for some scalar } c, \text{ so that } \quad P = LKP\ .$$

Because $L\mathbf{1} = \mathbf{0}$, $LP = L$, and so $P = LPKP$. From an easy check of the definitions, it then follows that $PKP = L^{\dagger}$, the *Moore-Penrose pseudoinverse* of L [**43**]. As a corollary, taking $G = \chi_n$, we find that $H = PKP$ shares the eigenvectors of L, reproducing the DCT.

3.3.1. Extension to Cartesian products of graphs.

Extending the complete result $H = L^\dagger$ from paths (or trees) to more general graphs is clearly not possible. For example, the graph distance matrix may be singular. However, obtaining the pseudoinverse of the Laplacian is not necessary – the appropriate discrete integral operator needs only to commute with the Laplacian, so as to have the same eigenvectors. The DCT, for example, is extended to higher dimensions via a tensor product construction, whose domain naturally corresponds to a Cartesian product of paths, which forms a grid, or lattice. In this setting, we can obtain a positive result.

First, an aside: from now on, whenever we refer to H or B for a weighted graph G, we always consider the shortest-path distances as computed on the graph $\widetilde{G}$ with the same edges as G, but whose edge weights are replaced by their reciprocals. Given that edge weights generally refer to *affinity* between vertices, these reciprocal weights instead look like a *repulsion* between vertices. When it is appropriate to think of the vertices as embedded in a metric space, edge weights must be a decreasing function of metric, while of course the reciprocal edge weights are an increasing function.

PROPOSITION 3.3.1. *Let G_1, G_2 be graphs with Laplacians L_1, L_2, and distance matrices B_1, B_2 respectively. Let $L_\times$ and $B_\times$ denote the graph Laplacian and distance matrix of the Cartesian product $G_1 \times G_2$. Define $H_i := P_i B_i P_i$ for $i \in \{1, 2, \times\}$, where P_i is the orthogonal projection on the orthogonal complement of $\mathbf{1}$ in the appropriate dimension. Then if H_1 commutes with L_1 and H_2 commutes with L_2, $H_\times$ commutes with $L_\times$.*

PROOF. The Laplacian of a Cartesian product of graphs may be written using the Kronecker sum/product as $L_\times = L_1 \oplus L_2 = L_1 \otimes I_2 + I_1 \otimes L_2$, where I_i ($i \in \{1, 2\}$) is the identity matrix of the appropriate size. The distance matrix of a Cartesian product of graphs may be written as $B_\times = B_1 \quad B_2 := B_1 \otimes J_2 + J_1 \otimes B_2$, where J_i ($i \in \{1, 2\}$) is the rank-one matrix $\mathbf{1}\mathbf{1}^\top$ of the appropriate size. The Kronecker product obeys the distributive rule $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$, whenever each product is defined. Some algebra reveals that then $H_\times = H_1 \quad H_2$. Hence, since

$L_1 J_1 = L_2 J_2 = \mathbf{O}$,

$$\begin{aligned} L_\times H_\times &= (L_1 \otimes I_2 + I_1 \otimes L_2)(H_1 \otimes J_2 + J_1 \otimes H_2) \\ &= L_1 H_1 \otimes J_2 + J_1 \otimes L_2 H_2 \\ &= L_1 H_1 \quad L_2 H_2 \\ &= H_1 L_1 \quad H_2 L_2 \\ &= H_\times L_\times \ , \end{aligned}$$

as desired. $\diamond$

This means that H reproduces the Laplacian eigenvectors of Cartesian products of weighted trees, and in particular for grid graphs, which makes these eigenvectors a candidate for extending the DCT to graphs. Particularly, the pseudo-inversion relationship for trees suggests that the appropriate analogue for the Fiedler vector $\boldsymbol{\phi}_1(L)$ of a general graph is then $\boldsymbol{\phi}_{n\ 1}(H)$; that is, the eigenvector corresponding to the top, or largest eigenvalue of H. First, recall L is positive semidefinite, so $\lambda_i(L) \geq 0$, $i = 0, \ldots, n \quad 1$. Now, the positive values $\lambda_i(L)$, have their order reversed after pseudoinversion, so the smallest positive value $\lambda_1(L)$ corresponds to the maximum value $\lambda_{n\ 1}(L^\dagger) = 1/\lambda_1(L)$. s one of the properties of the pseudoinverse $L^\dagger$ is that $LL^\dagger$ is the orthogonal projection against the kernel of L , and L is symmetric, then from the property $L^\dagger L L^\dagger = L^\dagger$ we conclude that $L\boldsymbol{x} = \implies L^\dagger \boldsymbol{x} =$, and so the eigenvalue $\lambda_0(L) = 0$ is reproduced, with $\lambda_0(L^\dagger) = 0$ as well. It is straightforward to show that this property holds for the Cartesian products in Propositions 3.3.1; while the order of the eigenvectors will vary in general, the extremal ones we use to partition do not.

PROPOSITION 3.3.2. *Let G_i, L_i, H_i be as in Proposition 3.3.1. Suppose that $\boldsymbol{\phi}_1(L_1) = \boldsymbol{\phi}_{n\ 1}(H_1)$ and $\boldsymbol{\phi}_1(L_2) = \boldsymbol{\phi}_{n\ 1}(L_2)$. Then, the eigenspace of $\lambda_1(L_\times)$ is the same as the eigenspace of $\lambda_{n\ 1}(H_\times)$.*

PROOF. First, given a matrix X, let $\widehat{X}$ denote the *vectorization* of X, a vector formed from the concatenation of the columns of X. Then for an $n \times n$ matrix A, an $m \times m$ matrix B, and $m \times n$ matrix X, we can concisely write matrix-vector multiplication for Kronecker products via $(A \otimes B)\widehat{X} = \widehat{BXA}$ **[42]**. This fact, with the form of $L_\times$, immediately recovers the well-known fact

that the eigenvalues of $L_\times$ are sums of the eigenvalues of L_1, L_2, and the eigenvectors of the former are Cartesian products of the latter. Setting $X_{ij} = \boldsymbol{\phi}_i(L_2)\boldsymbol{\phi}_j(L_1)$, we see that

$$L_\times \widehat{X_{ij}} = \widehat{X_{ij}L_1} + \widehat{L_2 X_{ij}} = \lambda_i(L_1)\widehat{X_{ij}} + \lambda_j(L_2)\widehat{X_{ij}} = (\lambda_i(L_1) + \lambda_j(L_2))\widehat{X_{ij}} \ .$$

There are nm such eigenvectors, providing a complete set. When the eigenvalues are unequal, they are orthogonal:

$$\widehat{X_{ij}} \ \ \widehat{X_{kl}} = \mathrm{tr}(\boldsymbol{\phi}_j(L_1)\boldsymbol{\phi}_i(L_2) \ \ \boldsymbol{\phi}_k(L_2)\boldsymbol{\phi}_l(L_1) \ \ ) = \delta_{ik}\delta_{jl} \ .$$

It must be that $\lambda_0(L_\times) = 0$, and $\boldsymbol{\phi}_0(L_\times) = \mathbf{1}$. Now, from Proposition 3.3.1, we know $H_\times$ shares these eigenvectors. However, noting that for each $i > 0$, $\boldsymbol{\phi}_i(L_1) \ \ \mathbf{1} = \boldsymbol{\phi}_i(L_2) \ \ \mathbf{1} = 0$,

$$H_\times \widehat{X_{ij}} = J\widehat{X_{ij}H_1} + \widehat{H_2 X_{ij}}J = (\delta_{i0}\lambda_j(H_1) + \delta_{j0}\lambda_i(H_2))\widehat{X_{ij}} \ .$$

This means that the only eigenvalues of $H_\times$ which aren't zero, are those for which exactly one of (i, j) in X_{ij} is 0. In other words, all mixed oscillations on the product graph have H-eigenvalue 0.

Now, first suppose $\lambda_1(L_1) \neq \lambda_1(L_2)$. Then $\lambda_1(L_\times) = \min(\lambda_1(L_1), \lambda_1(L_2))$ and $\boldsymbol{\phi}_1(L_\times)$ is either $\widehat{X_{1\,0}}$ or $\widehat{X_{0\,1}}$ accordingly. In either case, then we have by assumption that $\boldsymbol{\phi}_{n\ \ 1}(H_\times) = \boldsymbol{\phi}_1(L_\times)$ as desired.

If $\lambda_1(L_1) = \lambda_1(L_2)$, then the eigenspace for $\lambda_1(L_\times)$ is degenerate, including one oscillation in each direction of the product graph, and it is spanned by $\widehat{X_{1\,0}}$ and $\widehat{X_{0\,1}}$. gain, by assumption, we have that these vectors are both the eigenvectors of $H_\times$ corresponding to eigenvalue $\lambda_{n\ \ 1}(H_\times)$. $\diamond$

We learn two things from the Cartesian case: *we should expect the top eigenvectors of the discrete integral operator to be very informative of the graph, and that the bottom eigenvectors may be totally uninformative.* The Cartesian case is extreme, where $(m \quad 1)(n \quad 1) + 1$ out of mn total eigenvalues are all 0, leading to an extremely degenerate eigenspace.

3.3.2. Partitioning and embedding via discrete integral operators.

s a first trivial example, consider the case of G_{c} with uniform weights, so $W = J \quad I$. Then $L = nI \quad J = nP$, and the Fiedler vector is degenerate over all oscillatory vectors; as one would expect, any partition is equivalent on this highly-symmetric graph. Because the shortest path between any pair of vertices is the existing edge, then $B = W = J \quad I$, so $H = \quad \frac{1}{2}P(J \quad I)P = \frac{1}{2}P$.

The Fiedler vector of H, its top eigenvector, is again degenerate over all oscillatory vectors, agreeing with that of L.

In the general case of G_c with arbitrary weights, $B = W^{\circ}$ and $H = \frac{1}{2}PW^{\circ}P$. For example, in the Laplacian Eigenmaps algorithm [**8**], a graph is constructed from samples of a manifold using a Gaussian kernel, where if the Euclidean distance between samples i, j is $d(x_i, x_j)$, their edge weight, when the edge is present, is $w_{ij} = \exp(-d(x_i, x_j)^2/\sigma)$ for some scale parameter $\sigma > 0$. In that case, we would have $[B]_{ij} = \exp(d(x_i, x_j)^2/\sigma)$.

In order to build intuition, we want to emphasize that the matrix B is the graph distance matrix with respect to the weights W°, not W itself, and except in the case of uniform weights, is thus *not* finding shortest paths according to number of edges in the path, or minimizing the sum of weights. Suppose $w_1, \ldots, w_k$ are the weights of some edges forming a path between v_i, v_j. The harmonic sum of these weights is defined by $h(w_1, \ldots, w_k) \coloneqq \left(\sum_i w_i^{-1}\right)^{-1}$. Then, the objective of the optimization in B is to *maximize the harmonic sum of weights* among possible paths, since $[B]_{ij} = \min h(w_1, \ldots, w_k)^{-1} = (\max h(w_1, \ldots, w_k))^{-1}$, where the optimization is over paths connecting v_i, v_j. The harmonic sum is known to be dominated by its minimum input. Let $w_m = \min_i w_i$, and notice that

$$\frac{1}{w_m} \leq \sum_i \frac{1}{w_i} \leq \frac{n}{w_m} ,$$

so that the distance for a potential shortest path will be dominated by the contribution of its smallest weight. Hence the calculation of B avoids edges with small weights, and when forced to use them, assigns large distances to the resulting paths.

Now, we'll study some relevant properties of H. First, it is clear that $\phi_{n-1}(H)$ corresponds to a positive eigenvalue, because $\operatorname{tr} H > 0$, as follows. For any M satisfying $[M]_{ij} \geq 0$, $[M]_{ii} = 0$ for each i, j, we have that

$$\sum_i [PMP]_{ii} = \sum_{ijk} [P]_{ij}[M]_{jk}[P]_{ki} = \sum_{jk} \left[P^2\right]_{jk} [M]_{jk} = \sum_{jk} [P]_{jk}[M]_{jk} = -\frac{1}{n}\sum_{j\neq k}[M]_{jk} < 0 .$$

As this condition is satisfied by B, we have $\operatorname{tr} H > 0$. If we had that *every* eigenvalue of H were non-negative, then this turns out to be equivalent to W° being a Euclidean distance matrix [**6**]. As an aside, because $L^{\dagger}$ is positive semidefinite with $L^{\dagger}\mathbf{1} = 0$, it has the form $-PCP$, with C a

Euclidean distance matrix called the *commute time* distance matrix, which expresses for vertices u, v the expected time for a random walk starting at u to reach v, then u again. In **[97]**, it is shown that under mild assumptions, for many classes of graphs, the commute time distance for a large graph reflects mainly local degree information, rather than connectivity, and so is not suitable for use in machine learning problems. While for trees, commute time distance and shortest-path distance are identical, the similarities end there, and the discrete integral operator shouldn't suffer these shortcomings.

Now, with the additional assumption that G is connected, B is irreducible[3], and so the Perron-Frobenius theorem holds **[6]**, and $\boldsymbol{\phi}_{n\ 1}(B)$ is all positive. Further, we can show that this is *not* the case for H; $\boldsymbol{\phi}_{n\ 1}(H)$ makes an excellent Fiedler vector in part because it *must* be oscillatory. Let $\theta > \max_{ij}[B]_{ij}$, and notice that $H = \ \frac{1}{2}PBP = \frac{1}{2}P(\theta J \quad B)P$, so that contrary to initial appearance, the projections sandwich a matrix with positive entries. Let $A \coloneqq \theta J \quad B$ be that matrix, and consider $R(\boldsymbol{x}) \coloneqq \boldsymbol{x}\ PAP\boldsymbol{x} = (P\boldsymbol{x})\ A(P\boldsymbol{x})$. Suppose $\boldsymbol{x} = \arg\max_{\{\boldsymbol{x}':\|\boldsymbol{x}'\|=1\}} R(\boldsymbol{x}')$. But then consider

$$R\left(\frac{P\boldsymbol{x}}{\|P\boldsymbol{x}\|}\right) = \frac{1}{\|P\boldsymbol{x}\|^2}R(\boldsymbol{x}) = \left(1 \quad \frac{(\mathbf{1}\ \boldsymbol{x})^2}{n}\right)^{\frac{1}{2}} R(\boldsymbol{x}) \ .$$

If $\boldsymbol{x} \neq P\boldsymbol{x}$ such that $\mathbf{1}\ \boldsymbol{x} \neq 0$, then $R\left(\frac{P\boldsymbol{x}}{\|P\boldsymbol{x}\|}\right) > R(\boldsymbol{x})$, a contradiction. Hence $\phi_{n\ 1}(H)$ must be oscillatory, and take on both positive and negative values.

In the spirit of Ratio Cut (2.16), what combinatorial problem does this Fiedler vector provide a relaxed solution for? Using the usual Rayleigh criterion and the above, we know that

$$\boldsymbol{\phi}_{n\ 1}(H) = \underset{\|\boldsymbol{x}\|=1}{\arg\min}\, \boldsymbol{x}\ PBP\boldsymbol{x} = \underset{\|\boldsymbol{x}\|=1, \boldsymbol{x}\perp}{\arg\min}\ \boldsymbol{x}\ B\boldsymbol{x} \ .$$

Let S denote a subset of $V(G)$, and $\overline{S}$ its complement in V. Let $\mathrm{Bvol}(S) \coloneqq \sum_{i,j\in S}[B]_{ij}$ denote the B-volume of S, or the sum of all distances between vertices in S. Define the *Distance Ratio Cut* as

$$\mathrm{Bcut}(S) \coloneqq \frac{1}{|S|}\,\mathrm{Bvol}(S) + \frac{1}{|\overline{S}|}\,\mathrm{Bvol}(\overline{S}) \ . \tag{3.4}$$

[3]If G is not connected, in fact B contains infinite values! This is addressed shortly in Subsection 3.3.4.

Then, a straightforward argument in the spirit of **[38]** shows that if $\boldsymbol{x}_S$ is piecewise-constant on S and $\overline{S}$, and satisfies $\|\boldsymbol{x}\| = 1, \boldsymbol{x} \perp \mathbf{1}$, then $\boldsymbol{x}_S B \boldsymbol{x}_S = \text{Bcut}(S) + c$, with c a constant independent of S. Hence, the combinatorial problem of selecting S to minimize $\text{Bcut}(S)$ relaxes to computing the sign of the Fiedler vector $\boldsymbol{\phi}_{n\ \ 1}(H)$. We expand on this form of argument in detail in the proof of Proposition 3.3.5.

What kind of partition minimizes Bcut? There is a simultaneous optimization for balancing the sizes of the two halves of the partition, and for as much B-volume of the graph as possible to be outside of either partition; i.e., to arrange so vertices with a large distance between them are in opposite halves of the partition. If the graph is weighted, from the earlier discussion, this means vertices whose shortest path contains an edge with small weight are discouraged from being on the same side of the partition.

Recent work **[90]** suggests that, except for relatively rare examples, whose characterization is yet-unknown, a graph distance matrix B will tend to have *nearly-constant* top eigenvector $\boldsymbol{\phi}_{n\ \ 1}(B)$, with all other eigenvalues $\lambda_i(B) < 0$, $i < n$; we will call such graph distance matrices *generic*. Reversing the order of these for B, we see that for such graphs, the nearly-constant bottom eigenvector $\boldsymbol{\phi}_0(B)$ with $\lambda_0(B) < 0$ serves as the usual DC component $\boldsymbol{\phi}_0(L) \propto \mathbf{1}$. Putting these together, a generic graph distance matrix B is then additionally a Euclidean distance matrix precisely when $\mathbf{1} \in \text{span}\{\boldsymbol{\phi}_0(B), \boldsymbol{\phi}_1(B), \ldots, \boldsymbol{\phi}_k(B)\}$, where k is the multiplicity of the eigenvalue 0. s an intuition, the reason we use the *top* eigenvector $\boldsymbol{\phi}_{n\ \ 1}(H)$ instead of the *second-smallest* eigenvector $\boldsymbol{\phi}_1(H)$ is that the underlying distance matrix uses the reciprocal weights, which is critical. Finally, it is useful to notice that the conceptual mapping between Euclidean (i.e., continuous) distance and graph distance involves a numerically and computationally small, but analytically nebulous and difficult leap, just as occurred when applying midpoint quadrature to obtain the discrete integral operators from the continuous analogue.

In order to construct a spectral embedding with H, continuing this reasoning, the d coordinates of vertex i are $([\boldsymbol{\phi}_{n\ \ 1}(H)]_i, ..., [\boldsymbol{\phi}_{n\ \ d}(H)]_i)$, the top eigenvectors of H. When the embedding is one-dimensional, this reverts to the value of the Fiedler vector, and when our goal is to bipartition the data, we further revert to using only the sign of the Fiedler vector.

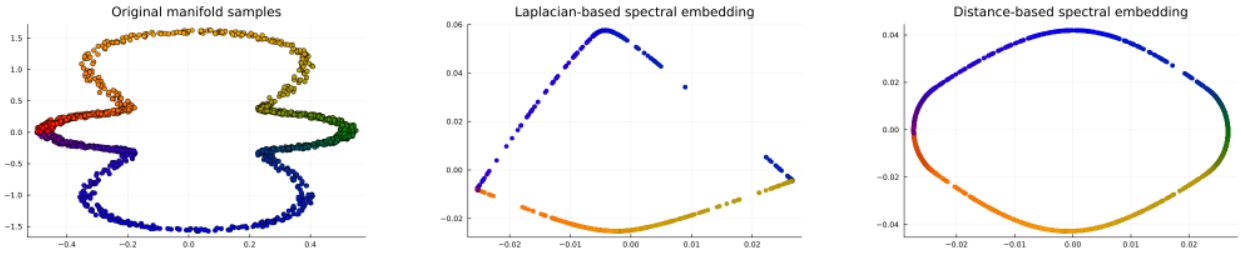


FIGURE 3.2. demonstration of the similar basic capabilities of the Laplacian and discrete integral operator-based spectral embeddings. On the left is noisy samples of a manifold, and on the right, two-dimensional spectral embeddings utilizing appropriate eigenvectors of the Laplacian and discrete integral operator, respectively.

Figure 3.2 demonstrates the similar basic capabilities of the Laplacian and discrete integral operator-based spectral embeddings. On the left is a scatterplot of 2000 noisy samples from a closed one-dimensional manifold given by the curve $y^2 = (x^2 + 0.1)(4 \quad x^2)$. The standard Gaussian kernel technique is used to generate an adjacency matrix from this scattered data [**8**]. Next are spectral embeddings, using two eigenvectors of each respective matrix. For the Laplacian, $\boldsymbol{\phi}_1(L)$ and $\boldsymbol{\phi}_2(L)$ are plotted – beginning with the Fiedler vector and in order of increasing eigenvalue. For the discrete integral operator H, $\boldsymbol{\phi}_{n \ 1}(H)$ and $\boldsymbol{\phi}_{n \ 2}(H)$ are used – beginning with the Fiedler vector and in order of decreasing eigenvalue. The marker colors are given by the polar angle of the original samples, and show that both embeddings have learned a smooth representation of the angle, and in fact, the distance-embedding appears higher quality. More advanced techniques (e.g., [**55**, **59**]) can capture finer details of the manifold, but higher-quality baselines suggest better performance with those techniques as well.

3.3.3. Normalized discrete integral operator. What is the equivalent, for a discrete integral operator, to the symmetric normalization often applied to the Laplacian to achieve improved embedding, partitioning, and other desirable spectral properties? For inspiration, we can return to Theorem 3.3.1, and the conclusion $H = L^{\dagger}$, to understand $(L^{\text{sym}})^{\dagger}$ in the case of trees. For any nonzero, non-orthogonal vectors $\boldsymbol{u}, \boldsymbol{v}$, let $P_{\boldsymbol{u}} = I \quad \frac{1}{\boldsymbol{u} \ \boldsymbol{u}} \boldsymbol{u}\boldsymbol{u}$ be the orthogonal projection against $\boldsymbol{u}$, and let $P_{\boldsymbol{u}\boldsymbol{v}} = I \quad \frac{1}{\boldsymbol{u} \ \boldsymbol{v}} \boldsymbol{u}\boldsymbol{v}$ be the oblique projection against $\boldsymbol{u}$, along $\boldsymbol{v}$.

PROPOSITION 3.3.3. *Let G be a tree, $W(G)$ its weighted adjacency matrix, and let L^{sym} be the symmetrically-normalized Laplacian of G. To avoid ambiguity with the usual degree vector $\boldsymbol{d}$ and*

reduce clutter from powers, let $\boldsymbol{\omega} \coloneqq (W\mathbf{1})^{\circ\frac{1}{2}} = \boldsymbol{d}^{\circ\frac{1}{2}}$, *and* $\Omega \coloneqq \mathrm{diag}(\boldsymbol{\omega})$. *Let* $K^{\mathrm{sym}} \coloneqq \frac{1}{2}\Omega B\Omega = \Omega K\Omega$ *be the symmetrically-normalized harmonic kernel matrix, as usual calculated using the weights in* W°. *Let* $H^{\mathrm{sym}} \coloneqq P_{\boldsymbol{\omega}} K^{\mathrm{sym}} P_{\boldsymbol{\omega}}$ *be the symmetrically-normalized discrete integral operator. Then,* $H^{\mathrm{sym}} = L^{\mathrm{sym}\dagger}$.

Proof. We begin with what we know: $K^{-1} = L - cI$ for a scalar c. So $L^{\mathrm{sym}} = \Omega^{-1} L \Omega^{-1}$, multiplying on both sides by Ω^{-1} will introduce the relevant terms, and we find

$$(K^{\mathrm{sym}})^{-1} = L^{\mathrm{sym}} - c(\Omega^{-1})(\Omega^{-1}) \quad .$$

Proceeding as before, notice that

$$K^{\mathrm{sym}}\Omega^{-1} = \Omega K = c'\Omega\mathbf{1} = c'\boldsymbol{\omega} \ ,$$

for another scalar c', and so multiplying on the left by K^{sym} yields $I = K^{\mathrm{sym}}L^{\mathrm{sym}} - cc'\boldsymbol{\omega}(\Omega^{-1}) \ .$ Multiplying on the left by $P_{\boldsymbol{\omega}}$ will kill the rank-one term, and notice that $P_{\boldsymbol{\omega}}L^{\mathrm{sym}} = L^{\mathrm{sym}} = L^{\mathrm{sym}}P_{\boldsymbol{\omega}}$. The same arguments apply for multiplying on the right, and we obtain $P_{\boldsymbol{\omega}}K^{\mathrm{sym}}P_{\boldsymbol{\omega}}L^{\mathrm{sym}} = H^{\mathrm{sym}}L^{\mathrm{sym}} = P_{\boldsymbol{\omega}} = L^{\mathrm{sym}}H^{\mathrm{sym}}$, yielding $H^{\mathrm{sym}} = L^{\mathrm{sym}\dagger}$ as desired. $\diamond$

We can also represent H^{sym} explicitly as a normalized version of H, using the oblique projection $P_{\boldsymbol{d}} \coloneqq I - \frac{1}{\boldsymbol{d}\ \mathbf{1}}\boldsymbol{d}\mathbf{1}$, by observing:

$$\Omega P_{\boldsymbol{\omega}} = \Omega - \frac{1}{\boldsymbol{d}\ \mathbf{1}}\boldsymbol{d}\boldsymbol{\omega} = P_{\boldsymbol{d}}\ \Omega \ ,$$

and that $PP_{\boldsymbol{d}} = P_{\boldsymbol{d}}$. Hence, applying the identity twice,

$$H^{\mathrm{sym}} = \Omega P_{\boldsymbol{d}}\ KP_{\boldsymbol{d}}\ \Omega = \Omega P_{\boldsymbol{d}}\ (PKP)P_{\boldsymbol{d}}\ \Omega = \Omega P_{\boldsymbol{d}}\ HP_{\boldsymbol{d}}\ \Omega = P_{\boldsymbol{\omega}}\Omega H\Omega P_{\boldsymbol{\omega}} \ .$$

In other words, the same "sandwich" used to form H^{sym} from B is the one used to form H^{sym} from H, in analogy with L^{sym}, W, and L respectively.

Unsurprisingly, by analogous arguments we can form $H^{\mathrm{rw}} \coloneqq PKDP_{\boldsymbol{d}}$, find that it satisfies $H^{\mathrm{rw}} = (L^{\mathrm{rw}})^{\dagger}$ for trees, and find that $H^{\mathrm{rw}} = HDP_{\boldsymbol{d}}$. Further, the relationship between the eigenvectors of H^{sym} and H^{rw} continues the metaphor with L.

PROPOSITION 3.3.4. *$(\lambda, \boldsymbol{\phi})$ is an eigenpair of H^{rw} if and only if $(\lambda, \Omega\boldsymbol{\phi})$ is an eigenpair of H^{sym}.*

PROOF. We know $(0, \boldsymbol{d})$ is an eigenpair for H^{rw}, and otherwise, $\boldsymbol{\phi}(H^{\text{rw}}) \perp \boldsymbol{d}$. Similarly, we know $(0, \boldsymbol{\omega})$ is an eigenpair for H^{sym}, and otherwise, $\boldsymbol{\phi}(H^{\text{sym}}) \perp \boldsymbol{\omega}$. Critically, this means that $P_{\boldsymbol{d}} H^{\text{rw}} = H^{\text{rw}}$. To see this, let Φ denote the $n \times (n \quad 1)$ matrix with $\boldsymbol{\phi}_1(H^{\text{rw}}), \ldots, \boldsymbol{\phi}_{n \ 1}(H^{\text{rw}})$ as columns, taking $\boldsymbol{\phi}_0(H^{\text{rw}}) = \boldsymbol{d}$ and Λ the $(n \quad 1) \times (n \quad 1)$ diagonal matrix of corresponding eigenvalues $\lambda_1(H^{\text{rw}}), \ldots, \lambda_{n \ 1}(H^{\text{rw}})$, for this proposition only, not necessarily in non-decreasing order. Then the spectral theorem yields

$$\begin{aligned} P_{\boldsymbol{d}} H^{\text{rw}} &= P_{\boldsymbol{d}} \begin{bmatrix} \boldsymbol{d} & \Phi \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{d} & \Phi \end{bmatrix}^{\ 1} \\ &= P_{\boldsymbol{d}} \begin{bmatrix} & \Phi \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{d} & \Phi \end{bmatrix}^{\ 1} \\ &= \begin{bmatrix} & \Phi \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{d} & \Phi \end{bmatrix}^{\ 1} = H^{\text{rw}} \ . \end{aligned}$$

Notice that, $\boldsymbol{\phi}_0(H^{\text{rw}}) = \boldsymbol{d} = \Omega\boldsymbol{\omega} = \Omega\boldsymbol{\phi}_0(H^{\text{sym}})$. Now, suppose $(\lambda, \boldsymbol{\phi})$ is any other eigenpair of H^{rw}. Then

$$HDP_{\boldsymbol{d}}\boldsymbol{\phi} = HD\boldsymbol{\phi} = \lambda\boldsymbol{\phi} \quad \text{, and so} \quad \Omega H\Omega(\Omega\boldsymbol{\phi}) = \lambda(\Omega\boldsymbol{\phi}) \ .$$

Let $\boldsymbol{\psi} = \Omega\boldsymbol{\phi}$, and notice $\boldsymbol{\psi} \perp \boldsymbol{\omega}$. Then $\Omega H\Omega\boldsymbol{\psi} = \Omega H\Omega P_{\boldsymbol{\omega}}\boldsymbol{\psi}$, and because this is itself proportional to $\boldsymbol{\psi}$,

$$\lambda\boldsymbol{\psi} = P_{\boldsymbol{\omega}}\Omega H\Omega P_{\boldsymbol{\omega}}\boldsymbol{\psi} = H^{\text{sym}}\boldsymbol{\psi} \ .$$

Each step is reversible, yielding the other direction as well. $\diamond$

Figure 3.3 repeats the embedding example from Figure 3.2, using the symmetric and random-walk normalization for each of the Laplacian-based and discrete integral operator-based spectral embeddings. In both cases, the data and technique are identical, only with L replaced by L^{sym}, L^{rw} and H replaced by $H^{\text{sym}}, H^{\text{rw}}$. Notice that they all do a more precise job of characterizing the underlying data, in different ways. The L-based embeddings produce more orthogonal embedding coordinates, and privilege a small slice of the total polar angle parameter space with the variance in $\boldsymbol{\phi}_2$. The H-based embeddings begin to correctly reproduce the geometry of the manifold, in addition to correctly reproducing the distribution of polar angles. theme which we have noticed

repeating is the relative fragility of the Laplacian-based embedding to noise and local changes in sampling density; distance-based embedding tends to identify underlying structure more robustly.

Finally, we define the *Distance Normalized Cut* $\mathrm{BNcut}(S)$ for $S \subset V$. Let $d_S \coloneqq \sum_{i \in S} [\boldsymbol{d}]_i$. Then

$$\mathrm{BNcut}(S) \coloneqq \frac{\mathrm{Bcut}(S) \quad \frac{1}{|V|}\mathrm{Bvol}(V)}{1 \quad \frac{1}{|V|\|\boldsymbol{d}\|^2}\left(\sqrt{\frac{|\overline{S}|}{|S|}}d_S \quad \sqrt{\frac{|S|}{|\overline{S}|}}d_{\overline{S}}\right)^2} . \tag{3.5}$$

Notice that minimizing $\mathrm{BNcut}(S)$, with the constraint that neither $S, \overline{S}$ is empty, attempts to simultaneously minimize $\mathrm{Bcut}(S)$, and minimize $\left(\sqrt{\frac{|\overline{S}|}{|S|}}d_S \quad \sqrt{\frac{|S|}{|\overline{S}|}}d_{\overline{S}}\right)^2$, which is achieved when possible by matching the ratios of degrees to cardinalities, such that $\frac{d_S}{d_{\overline{S}}} = \frac{|S|}{|\overline{S}|}$. This second influence acts as a regularizer, and punishes Bcut optimizers which choose extreme solutions, like isolating a few high-degree vertices into a partition.

Fulfilling the metaphor with L, the sign of the Fiedler vector of H^{rw} solves a relaxed version of the problem $\min_{\emptyset \subsetneq S \subsetneq V} \mathrm{BNcut}(S)$.

Proposition 3.3.5. *First,*

$$\boldsymbol{\phi}_{n \ \ 1}(H^{\mathrm{rw}}) = \underset{: \|P \quad \|=1, \quad =0}{\arg\min} \quad \boldsymbol{\phi} \ \ B\boldsymbol{\phi} .$$

Next, for $S \subset V$, let $\boldsymbol{x}_S$ be piecewise constant on $S, \overline{S}$, satisfying $\|P_{\boldsymbol{d}}\boldsymbol{x}_S\| = 1, \boldsymbol{x}_S\mathbf{1} = 0$. Then

$$\boldsymbol{x}_S B \boldsymbol{x}_S = \mathrm{BNcut}(S) .$$

Proof. Let $\boldsymbol{x} = \boldsymbol{\phi}_{n \ \ 1}(H^{\mathrm{rw}}), \lambda = \lambda_{n \ \ 1}(H^{\mathrm{rw}})$. Then $PKDP_{\boldsymbol{d}}\boldsymbol{x} = \lambda \boldsymbol{x}$, and so

$$P_{\boldsymbol{d}}DPKDP_{\boldsymbol{d}}\boldsymbol{x} = \lambda P_{\boldsymbol{d}}D\boldsymbol{x} .$$

Because the projections satisfy $P_{\boldsymbol{d}}DP = P_{\boldsymbol{d}}D$, the matrix on the left is symmetric, and so

$$\boldsymbol{x} = \underset{: \|P \ D \ \|=1, \quad \boldsymbol{d}=0}{\arg\max} \quad \boldsymbol{\phi} \ \ P_{\boldsymbol{d}}DKDP_{\boldsymbol{d}}\boldsymbol{\phi} = \underset{: \|P \ D \ \|=1, \quad \boldsymbol{d}=0}{\arg\max} \quad \boldsymbol{\phi} \ \ DKD\boldsymbol{\phi} .$$

Replacing $\boldsymbol{\phi}$ with $D^{-1}\boldsymbol{\phi}$, and noting $B = -2K$,

$$\boldsymbol{x} = \underset{\boldsymbol{\phi} : \boldsymbol{\phi}^T D \mathbf{1} = 0, \; \boldsymbol{\phi}^T D \boldsymbol{\phi} = 1}{\arg\min} \boldsymbol{\phi}^T B \boldsymbol{\phi} \; ,$$

as desired. Next, let $\emptyset \subsetneq S \subsetneq V$, and let $[\boldsymbol{x}_S]_i = a\delta_{i \in S} + b\delta_{i \in \overline{S}}$. Then

$$\boldsymbol{x}_S^T B \boldsymbol{x}_S = (a^2 - ab)\,\mathrm{Bvol}(S) + (b^2 - ab)\,\mathrm{Bvol}(\overline{S}) + ab\,\mathrm{Bvol}(V) \; .$$

The values of a, b are determined by the constraints on $\boldsymbol{x}_S$. From $\boldsymbol{x}_S^T \mathbf{1} = 0$, we find $b = -\frac{|S|}{|\overline{S}|}a$. Next,

$$\begin{aligned}
\|P_{\boldsymbol{d}} \boldsymbol{x}_S\|^2 = \boldsymbol{x}_S^T P_{\boldsymbol{d}} \boldsymbol{x}_S &= \|x_S\|^2 - \frac{(\boldsymbol{x}_S^T \boldsymbol{d})^2}{\|\boldsymbol{d}\|^2} \\
&= a^2|S| + b^2|\overline{S}| - \frac{(ad_S + bd_{\overline{S}})^2}{\|\boldsymbol{d}\|^2} \\
&= a^2 \frac{|S||V|}{|\overline{S}|} \left[1 - \frac{1}{|V|\|\boldsymbol{d}\|^2} \left(\sqrt{\frac{|\overline{S}|}{|S|}} d_S - \sqrt{\frac{|S|}{|\overline{S}|}} d_{\overline{S}} \right)^2 \right] \\
&=: a^2 \frac{|S||V|}{|\overline{S}| F_S} \; .
\end{aligned}$$

The large bracketed expression is denoted F_S^{-1} in the last line. Now, the condition $\|P_{\boldsymbol{d}} \boldsymbol{x}_S\| = 1$ yields

$$a^2 = \frac{|\overline{S}| F_S}{|S||V|} \; , \; b^2 = \frac{|S| F_S}{|\overline{S}||V|} \; , \; ab = -\frac{F_S}{|V|} \; ,$$

and so

$$\begin{aligned}
\boldsymbol{x}_S^T B \boldsymbol{x}_S &= F_S \left(\frac{\mathrm{Bvol}(S)}{|S|} + \frac{\mathrm{Bvol}(\overline{S})}{|\overline{S}|} - \frac{\mathrm{Bvol}(V)}{|V|} \right) \\
&= F_S \left(\mathrm{Bcut}(S) - \frac{1}{|V|} \mathrm{Bvol}(V) \right) = \mathrm{BNcut}(S) \; ,
\end{aligned}$$

as desired. Hence, if we restrict the optimization problem for computing $\boldsymbol{\phi}_{n-1}(H^{\mathrm{rw}})$ to only piecewise-constant vectors on a bipartition, we exactly optimize BNcut. Equivalently, optimizing BNcut relaxes to computing the sign of the Fiedler vector. $\diamond$

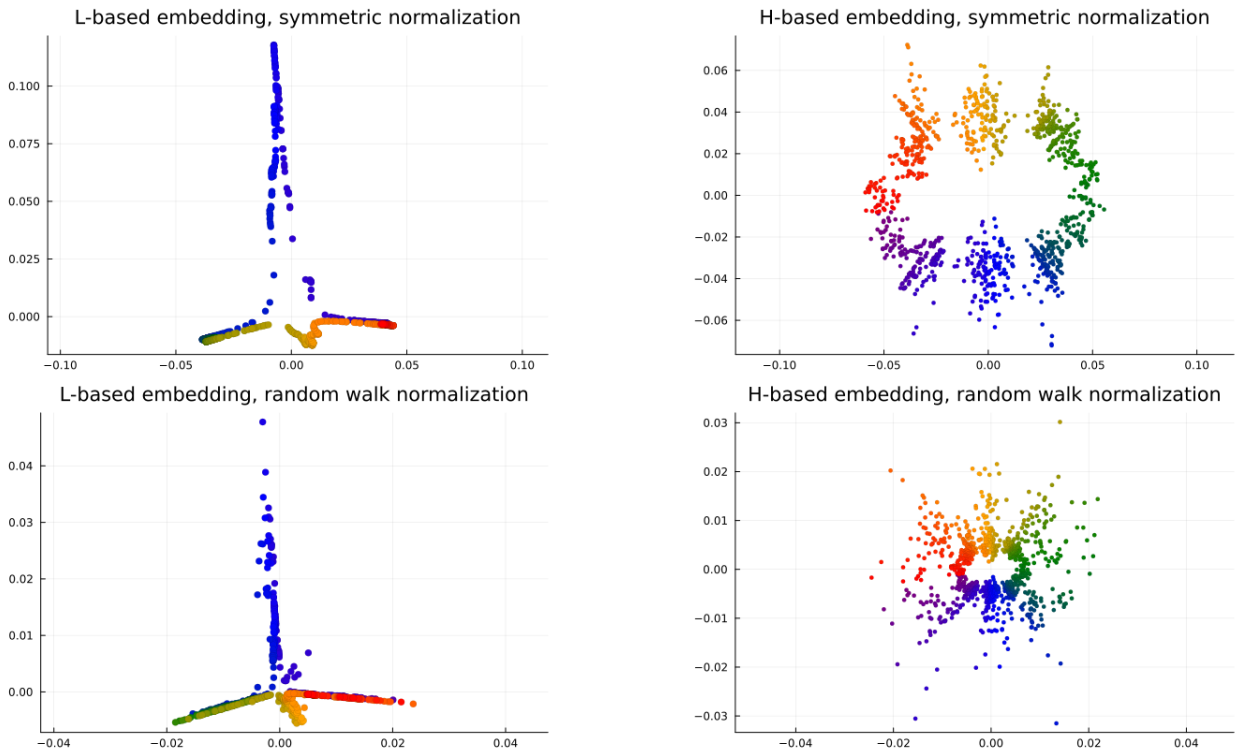


FIGURE 3.3. The embedding example from Figure 3.2, using the symmetric and random-walk normalization for each of the Laplacian-based and discrete integral operator-based spectral embeddings. In both cases, the data and technique are identical, only with L replaced by L^{sym}, L^{rw} and H replaced by $H^{\text{sym}}, H^{\text{rw}}$.

3.3.4. Censoring the distance matrix. While the embedding examples in Figures 3.2, 3.3 suggest the utility of the discrete integral operator in representing spectral geometry, there are two key problems that arise in its use:

- The Laplacian of a graph G is *sparse*, with only $\mathcal{O}(|V(G)| + |E(G)|)$ entries and so computing its Fiedler vectors is generally fast. The distance matrix of a connected graph is *dense*, requiring an expensive operation to compute it, dramatically more storage, and a lengthier Fiedler vector computation on the resulting discrete integral operator.
- If G is disconnected, worse yet, the distance matrix is populated with infinities between any two vertices for which there is no shortest path, and no Fiedler vector can be computed at all.

Regarding the first problem, the *Fast Multipole Method* can be used to accelerate the computation of the eigenvectors of the distance matrix, as was explored in the context of computing

the Laplacian eigenvectors of general shape Euclidean domains in **[101]**. While this is an attractive avenue to incorporate for future work, we instead address these problems theoretically and pragmatically by *censoring* large values of the distance matrix. Given a graph distance matrix B and $m \in \mathbb{R}_{>0}$, let $[B_m]_{ij} \coloneqq \min([B]_{ij}, m)$. Likewise, define $K_m, K_m^{\text{sym}}, H_m, H_m^{\text{sym}}$ as the respective counterpart dependent on B_m.

Implementing this one idea of censoring has many consequences:

- finite threshold removes all infinities from a distance matrix, re-enabling eigenvector calculation for disconnected graphs.
- Distances attaining the constant finite threshold need not be stored, recovering sparsity of the distance matrix dependent on the size of the threshold.
- Large distances (relative to others in the graph) are intuitively less meaningful for measuring structure of the graph, and as our experiments suggest, may even sometimes introduce undesired noise into geometry represented by a spectral embedding. Thus, censoring can act as a form of regularization.
- lgorithmically, it is possible to accelerate the computation of the distance matrix dramatically, as a sparse matrix, by making the threshold value known prior to computation.
- Even though H is generally dense, when B_m is sparse, we can use the fact that H is a low-rank perturbation of $\frac{1}{2}B_m$ to compute the embedding for our discrete integral operator with only a little more difficulty and expense than doing so for B_m.

Explicitly, we proceed in several steps to efficiently construct a discrete integral operator-based spectral embedding for a graph G from a censored distance matrix. First, given the adjacency matrix W for G, we construct the reciprocal weights W°, and define a censorship threshold m. Next, for each vertex in G, we run a modified version of Dijkstra's algorithm[4] **[22]**, in order to obtain for each other vertex, either the length of the shortest path to that vertex in W°-weights, if it doesn't exceed m, or else m, yielding $[B_m]_{ij}$ for each vertex pair (v_i, v_j). Critically, our modification does not require storing any values at or exceeding the threshold while computing the shortest path lengths, and does not require searching on paths whose length would exceed the threshold.

[4]We perform these runs sequentially, but runs of Dijkstra's algorithm initialized at distinct vertices are trivially parallelizable, so parallel runs here offer a trade-off of greater memory footprint for faster execution.

In order to accomplish this in code, we need to understand the relationship between the below-threshold and at-threshold entries of B_m. Let $T_m \coloneqq \{(i,j) \mid 0 < [B_m]_{ij} < m\}$ be the non-threshold support of B_m, and let $[\mathbf{1}_{T_m}]_{ij} \coloneqq \delta_{(i,j)\in T_m}$ be its indicator function. Let $[B_m^*]_{ij} \coloneqq [\mathbf{1}_{T_m}]_{ij}[B_m]_{ij}$ be the restriction of B_m to that support, whose nonzero entries are the non-thresholded values in B_m. B_m^* is precisely the matrix we will obtain the entries of by running the sparse `dijkstra` code from lgorithm 1, and clearly we have $B_m = B_m^* + m(J \quad I \quad \mathbf{1}_{T_m})$. Hence,

$$H_m = \frac{1}{2}PB_mP = \frac{1}{2}P(B_m^* + m(J \quad I \quad \mathbf{1}_{T_m}))P = \frac{1}{2}P(mI + m\mathbf{1}_{T_m} \quad B_m^*)P \ .$$

Examining each interior term, mI is diagonal, and the two other terms share the support of T_m, sparse and purely off-diagonal, taking the combined positive values $[m\mathbf{1}_{T_m} \quad B_m^*]_{ij} = m \quad [B_m^*]_{ij} > 0$. We call this matrix $[K_m^*]_{ij} \coloneqq \frac{1}{2}(m\delta_{ij} + m \quad [B_m^*]_{ij})$ the *censored harmonic kernel*, and it is exactly what we compute in order to obtain the discrete integral operator $H_m = PK_mP = PK_m^*P$. Likewise, easily following from the argument in Equation 3.3.3, let $K_m^{\mathrm{sym}*} = \Omega K_m^*\Omega$, and then $H_m^{\mathrm{sym}} = P_{\boldsymbol{\omega}}K_m^{\mathrm{sym}}P_{\boldsymbol{\omega}} = P_{\boldsymbol{\omega}}K_m^{\mathrm{sym}*}P_{\boldsymbol{\omega}}$.

s an example, recall in Subsection 3.3.2, when we assumed each $[B]_{ij} < \theta$; then $B_\theta^* = B$, $\mathbf{1}_{T_\theta} = J \quad I$, and in this case the censored harmonic kernel is $K_\theta^* = \frac{1}{2}(\theta J \quad B)$, and of course, its entries are positive. This illustrates how the harmonic kernel should be seen fundamentally as non-negative, because once the projections are applied in our calculation of H, we are always free to add a constant to every entry of B.

Next, consider censoring the distance matrix for the complete graph G_c at $m = \epsilon^{-1}$. Then

$$[K_m^*]_{ij} = \frac{1}{2}\left(\frac{1}{\epsilon}\delta_{ij} + \left[\frac{1}{\epsilon} \quad \exp(d_{ij}^2)\right]_+\right) = \frac{1}{2\epsilon}\left(\delta_{ij} + \exp(d_{ij}^2)[\exp(-d_{ij}^2) \quad \epsilon]_+\right) \ ,$$

which means we are effectively performing *soft thresholding* on the weights, just as is typically performed when constructing and sparsifying an adjacency matrix from scattered data[5].

Returning to the sparse computation, finally, we do not explicitly form H_m or H_m^{sym}; instead, as typical with eigensolver libraries like RP CK [**64**], we simply provide the `mul!` function (in Julia parlance) which performs in-place multiplication of H_m (or H_m^{sym}) by a vector. s Ω is diagonal,

[5]In fact, the instructions in many prominent papers such as [ ] recommend hard thresholding, likely for simplicity rather than an opinion on soft vs. hard thresholding.

$[K_m^{\text{sym}*}]_{ij} = [\boldsymbol{\omega}]_i[\boldsymbol{\omega}]_j[K_m^*]_{ij}$ has the same sparsity structure as K_m^*. Let $\boldsymbol{k} \coloneqq K_m^*\mathbf{1}$ and $\boldsymbol{\ell} \coloneqq K_m^{\text{sym}*}\boldsymbol{\omega}$. The necessary function can be easily determined by:

$$\boldsymbol{x} \mapsto H_m\boldsymbol{x} = K_m^*\boldsymbol{x} \quad \frac{1}{n}\left(\boldsymbol{k}\ \ \boldsymbol{x} \quad \frac{(\mathbf{1}\ \ \boldsymbol{k})\ (\mathbf{1}\ \ \boldsymbol{x})}{n}\right)\mathbf{1} \quad \frac{\mathbf{1}\ \ \boldsymbol{x}}{n}\boldsymbol{k} \ , \text{ and}$$

$$\boldsymbol{x} \mapsto H_m^{\text{sym}}\boldsymbol{x} = K_m^{\text{sym}*}\boldsymbol{x} \quad \frac{1}{\boldsymbol{\omega}\ \ \boldsymbol{\omega}}\left(\boldsymbol{\ell}\ \ \boldsymbol{x} \quad \frac{(\boldsymbol{\omega}\ \ \boldsymbol{\ell})\ (\boldsymbol{\omega}\ \ \boldsymbol{x})}{\boldsymbol{\omega}\ \ \boldsymbol{\omega}}\right)\boldsymbol{\omega} \quad \frac{\boldsymbol{\omega}\ \ \boldsymbol{x}}{\boldsymbol{\omega}\ \ \boldsymbol{g}}\boldsymbol{\ell} \ .$$

With these steps in place, we are able to calculate discrete integral operator-based embeddings with often only a small multiple in resource usage and execution time over an equivalent Laplacian embedding, even though this embedding appears to capture far more structure of the graph. In lgorithm 1, we detail the modified version of Dijkstra's algorithm, with our explicit Julia code, and include the `distmat` and `censored_distmat` sparse matrix constructors alongside it. In Figure 3.4, we demonstrate the power of the new embeddings to solve a problem which is often tricky, if not impossible, for the equivalent Laplacian implementation. The goal is to separate two noisily-sampled, thick spiral bands of the form $r = a\theta + b$, by constructing the usual Gaussian kernel from Euclidean distance and partitioning the resulting graph. bandwidth of $\frac{\pi}{a}$ would cause the spirals to merge into a single annular shape, and make classification impossible. "difficult" regime begins at $0.5\frac{\pi}{a}$, after which the Laplacian Fiedler vector will generally only recover the true classes with significant fine-tuning of the kernel parameters, if at all. For this experiment, we generate 1500 points, with parameters $a = \frac{1}{2\pi}, b = 1$, and bandwidth $0.6\frac{\pi}{a}$. The top-left spiral on the right is a typical result of Laplacian classification, where blue and red indicate the classification (i.e., the sign of the Fiedler vector). Conversely, we find that the integral operator-based classification is much more robust, able to reliably reproduce the planted classification for a wide range of censorship thresholds (and sometimes with no censorship required), far deeper into the difficult regime than for the Laplacian. Here we performed 11 further classification attempts, using the Fiedler vector of H_m, censored at the values in `[400, 350, 300, 280, 250, 210, 170, 130, 90, 60, 30]`, which are chosen to span the notable features in the histogram at top-left. This histogram illustrates the distribution of distance values contained in B, and the grey dashed lines, read right-to-left, indicate successively smaller thresholds applied to H. The bottom-left chart shows the fraction of shortest paths whose length is below each threshold, and so approximately gives the filling fraction

for the corresponding sparse B_m. Each classification is then illustrated in the spirals on the right, in descending threshold order, read left-to-right, top-to-bottom. Notice that leaving the extremely long shortest-paths in the graph actually damages the classification, and here, a perfect result is achieved between thresholds 130 and 300, covering approximately a third of the possible range of thresholds, and resulting in distance matrices with filling factor ranging from 20-65%. The equivalent Laplacian has a filling factor of about 5%, though it fails to achieve any meaningful classification. Further reduction of the threshold effectively reduces the graph to mostly-disconnected "dust", and classification performance degrades.

lgorithm 1 Dijkstra's lgorithm, Sparsified with Threshold (Julia)

```
fun tion dijkstra(g:: bstractGraph, src::Int; dmax = Inf)
    dists = sparsevec([src], [0.0], nv(g))

    H = PriorityQueue{Int,Float64}()
    H[src] = 0.0

    while !isempty(H)
        u = dequeue!(H)
        d = dists[u]
        for v in outneighbors(g, u)
            alt = min(d + weights(g)[u, v], dmax)
            if alt < dmax && (v ∉ findnz(dists)[1] || alt < dists[v])
                dists[v] = alt
                H[v] = alt
            end
        end
    end

    dists
end

distmat(g:: bstractGraph; dmax=Inf) = reduce(
    hcat,
    (dijkstra(g, i; dmax) for i ∈ 1:nv(g))
)

censored_distmat(g:: bstractGraph, dmax::Real) = reduce(
    hcat,
    (
        findnz(dists) |> nzds -> sparsevec(nzds[1], dmax .- nzds[2], nv(g))
        for dists ∈ (dijkstra(g, i; dmax) for i ∈ 1:nv(g))
    )
)
```

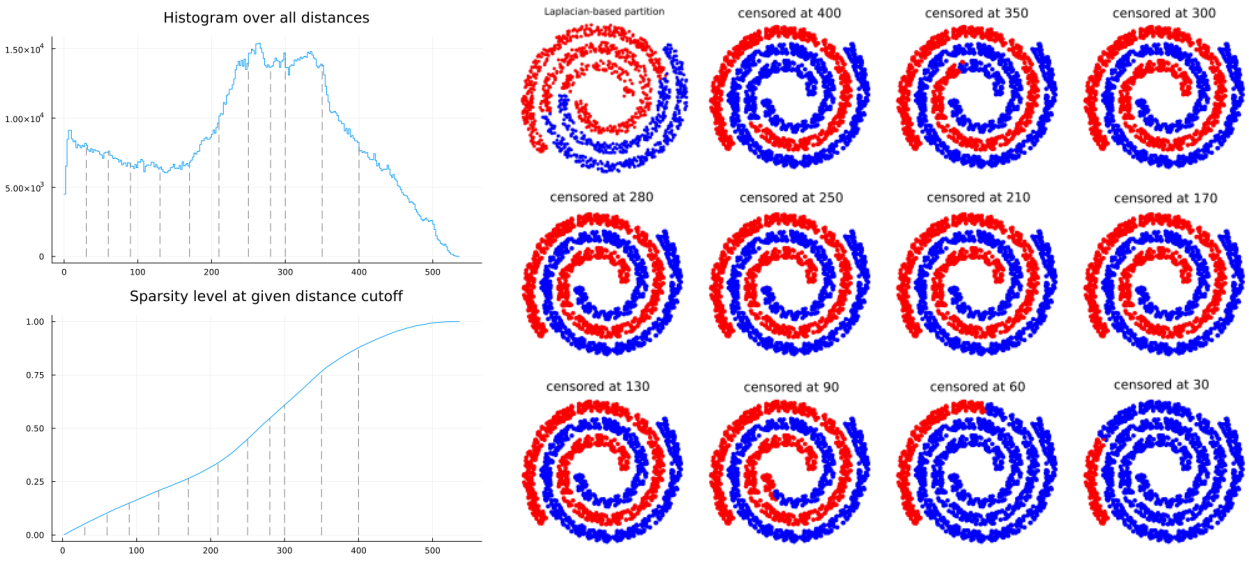


FIGURE 3.4. Results of an unsupervised classification experiment, with the goal of separating the two spirals pictured at right. The histogram at top-left illustrates the distribution of distance values contained in B. The classification is performed 12 times, once with the Laplacian, and 11 times with the censored distance matrix, and each grey dashed line in the histogram indicates the censorship threshold for a trial. The bottom-left chart shows the fraction of shortest paths whose length is below each threshold. Each classification is illustrated in the spirals on the right, first by the Laplacian, and then by the censored discrete integral operator, in order of decreasing threshold.

CH PTER 4

Recursive -Region Partitioning

Now, we turn to the key ingredients of this dissertation: developing a Fiedler vector suitable for partitioning κ-regions, and multiscale transforms suitable for representing functions on κ-regions.

4.1. Simplex Consistency and the Hodge Laplacian

Recall the boundary operators $[B_\kappa]_{\sigma} = \mathrm{rel}(\sigma, \alpha)$ introduced in Section 2.4. They represent discrete differential operators encoding the structure of some part of a simplicial complex, and so can be building blocks towards a spectral analysis of functions on that simplicial complex. For analyzing functions on κ-simplices with $\kappa > 0$, we will construct operators based on the *Hodge Laplacian*, or *κ-Laplacian*. Before considering weights, the *combinatorial* κ-Laplacian is defined for κ-simplices as

$$L_\kappa \coloneqq B_{\kappa\ 1} B_{\kappa\ 1} + B_\kappa B_\kappa \ .$$

We refer to the *lower* and *upper* κ-Laplacians as $L_\kappa^\vee \coloneqq B_{\kappa\ 1} B_{\kappa\ 1}$ and $L_\kappa^\wedge \coloneqq B_\kappa B_\kappa$, respectively.

For the rest of this section, let σ, τ be oriented κ-simplices. Recall our notation and definitions from Section 2.4, where when $\sigma \sim \tau$, we denote the boundary $\mathrm{bd}(\sigma, \tau)$ by α, and when $\sigma \simeq \tau$, we denote the hull $\mathrm{hl}(\sigma, \tau)$ by β.

Now, suppose $\sigma \sim \tau$. Then define the *consistency* of σ, τ by

$$\mathrm{con}(\sigma, \tau) \coloneqq \quad p_\sigma p_\tau \,\mathrm{nat}(\sigma, \alpha)\,\mathrm{nat}(\tau, \alpha) = \quad \mathrm{rel}(\sigma, \alpha)\,\mathrm{rel}(\tau, \alpha) \ .$$

When $\sigma \quad \tau$, we set $\mathrm{con}(\sigma, \tau) = 0$. We say σ, τ are *consistent* or have *consistent orientation* when $\mathrm{con}(\sigma, \tau)$ is positive, and *inconsistent* when the same is negative. Suppose, in addition, $\sigma \simeq \tau$. Then define the *hull consistency* of σ, τ by

$$\mathrm{hcon}(\sigma, \tau) \coloneqq \quad p_\sigma p_\tau \,\mathrm{nat}(\beta, \sigma)\,\mathrm{nat}(\beta, \tau) = \quad \mathrm{rel}(\beta, \sigma)\,\mathrm{rel}(\beta, \tau) \ .$$

FIGURE 4.1. Pairs of κ-simplices demonstrating consistency at their boundary face, for $\kappa = 1, 2$. The mixed-color pairs are consistent, and the same-color pairs are inconsistent.

Likewise, when $\sigma \not\supseteq \tau$, $\mathrm{hcon}(\sigma, \tau) = 0$. We say σ, τ are *hull-consistent* when $\mathrm{hcon}(\sigma, \tau)$ is positive, and *hull-inconsistent* when the same is negative.

Notice that if σ, τ are consistent, then $\alpha \in \sigma^{\mathrm{h}} \iff \alpha \in \tau^{\mathrm{t}}$, and vice versa. Likewise, if σ, τ are inconsistent, then $\alpha \in \sigma^{\mathrm{h}} \iff \alpha \in \tau^{\mathrm{h}}$, and vice versa. Hence, consistent simplices use their shared boundary face in opposite ways; one considering it a head face, and the other a tail face. Inconsistent simplices use their shared boundary face identically; both considering it either a head face or tail face. In the case of $\kappa = 1$, two directed edges are consistent when they *flow* into each other at their boundary vertex, and are inconsistent when they *collide* at the boundary vertex, either both pointing toward it, or both pointing away. Cases for $\kappa = 1, 2$ are demonstrated in Figure 4.1

key observation is that the notions of consistency and hull-consistency are exactly *opposite*; consistency with respect to the shared boundary is inconsistency with respect to the shared hull, and vice versa.

LEMMA 4.1.1. *Let $\kappa > 0$. For strongly adjacent oriented κ-simplices σ, τ, we have that*

$$\mathrm{con}(\sigma, \tau) = \quad \mathrm{hcon}(\sigma, \tau) \ .$$

PROOF. Let $\alpha = \mathrm{bd}(\sigma, \tau)$, and $\beta = \mathrm{hl}(\sigma, \tau)$. Then let $\ell = \iota(\beta \setminus \sigma)$ and $m = \iota(\beta \setminus \tau)$ be the positions in β given by (2.19).

First, suppose $\ell < m$. Then we must have that $\iota(\sigma \setminus \alpha) = m \quad 1$, and $\iota(\tau \setminus \alpha) = \ell$. Hence,

$$\begin{aligned} \mathrm{con}(\sigma, \tau) = &\quad p_\sigma p_\tau \,\mathrm{nat}(\sigma, \alpha)\,\mathrm{nat}(\tau, \alpha) \\ = &\quad p_\sigma p_\tau (\quad 1)^m (\quad 1)^{\ell+1} \\ = &\; p_\sigma p_\tau \,\mathrm{nat}(\beta, \tau)\,\mathrm{nat}(\beta, \sigma) \\ = &\quad \mathrm{hcon}(\sigma, \tau) \ , \end{aligned}$$

as desired. If instead $m < \ell$, then $\iota(\sigma \setminus \alpha) = m$, and $\iota(\tau \setminus \alpha) = \ell \quad 1$. From there the proof proceeds identically as in the $\ell < m$ case. $\diamond$

To illustrate this result, consider a naturally-oriented 2-simplex s, as in Figure 4.2, with naturally-oriented 1-faces e, f, g. Notice $e, f \in s^{\mathrm{h}}$, $g \in s^{\mathrm{t}}$. Checking each pair, (e, f) are hull-inconsistent but consistent, (e, g) are hull-consistent but inconsistent, and (f, g) are hull-consistent but inconsistent. From the definition it is clear that flipping the orientation of s cannot change the hull-consistency; here we see this is because the membership of $s^{\mathrm{h}}, s^{\mathrm{t}}$ themselves merely switch. Now, consider flipping the orientation of some edge, without loss of generality e. This will have two effects: to flip the sign of $\mathrm{hcon}(e, f)$ and $\mathrm{hcon}(e, g)$ directly, and to swap the membership of $e^{\mathrm{h}}, e^{\mathrm{t}}$, which will flip the sign of $\mathrm{con}(e, f)$ and $\mathrm{con}(e, g)$. To summarize, when edges *flow* into each other along their boundary vertex, and so use that vertex in *opposite* ways, they are in turn used by the triangle to whom they are faces in *identical* ways. Conversely, when edges *conflict* at their boundary vertex, and so use that vertex in *identical* ways, they are in turn used by the triangle to whom they are faces in *opposite* ways.

corollary of Lemma 4.1.1 is the well-known fact that the discrete codifferential, as a coboundary operator, satisfies $B_\kappa \;{}_1B_\kappa = \quad$ **[65]**. In other words, "the boundary of a boundary is empty".

OROLLARY 4.1.1. *For any oriented simplicial complex C and $\kappa > 0$,*

$$B_\kappa \;{}_1B_\kappa = \quad . \tag{4.1}$$

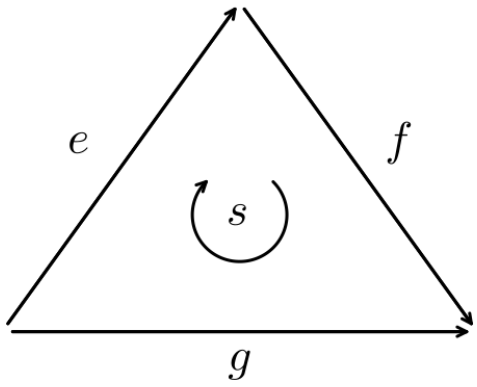


FIGURE 4.2. naturally-oriented 2-simplex s, with naturally-oriented 1-faces e, f, g.

PROOF. Let $\alpha \in C_{\kappa\ \ 1}, \beta \in C_{\kappa+1}$. Then

$$[B_{\kappa\ \ 1}B_\kappa]_{\ \beta} = \sum_{\sigma\in C_\kappa} [B_{\kappa\ \ 1}]_{\ \sigma}\,[B_\kappa]_{\sigma\beta} = p\ \ p_\beta \sum_{\sigma\in C_\kappa} \mathrm{nat}(\sigma,\alpha)\,\mathrm{nat}(\beta,\sigma)\ .$$

The final sum vanishes for any σ that does not satisfy $\alpha \subset \sigma \subset \beta$. If $\alpha \not\subset \beta$, no σ satisfies this condition and the sum vanishes, yielding the desired result. Otherwise, there are precisely two distinct κ-simplices σ_1, σ_2 satisfying this condition, and we have $\mathrm{bd}(\sigma_1,\sigma_2) = \alpha$, $\mathrm{hl}(\sigma_1,\sigma_2) = \beta$. By Lemma 4.1.1, and noting $\mathrm{nat}(\cdot,\cdot) \in \{\pm 1\}$,

$$\begin{aligned} 0 = \mathrm{con}(\sigma_1,\sigma_2) + \mathrm{hcon}(\sigma_1,\sigma_2) = \quad & p_\sigma p_\tau \left(\mathrm{nat}(\sigma_1,\alpha)\,\mathrm{nat}(\sigma_2,\alpha) + \mathrm{nat}(\beta,\sigma_1)\,\mathrm{nat}(\beta,\sigma_2)\right) \\ & = \mathrm{nat}(\sigma_1,\alpha)\,\mathrm{nat}(\sigma_2,\alpha) + \mathrm{nat}(\beta,\sigma_1)\,\mathrm{nat}(\beta,\sigma_2)\ . \end{aligned}$$

Hence $\mathrm{nat}(\sigma_2,\alpha) = \quad \mathrm{nat}(\sigma_1,\alpha)\,\mathrm{nat}(\beta,\sigma_1)\,\mathrm{nat}(\beta,\sigma_2)$, and we conclude that

$$\begin{aligned} [B_{\kappa\ \ 1}B_\kappa]_{\ \beta} &= p\ \ p_\beta \left(\mathrm{nat}(\sigma_1,\alpha)\,\mathrm{nat}(\beta,\sigma_1) + \mathrm{nat}(\sigma_2,\alpha)\,\mathrm{nat}(\beta,\sigma_2)\right) \\ &= p\ \ p_\beta\ \ \mathrm{nat}(\sigma_1,\alpha)\,\mathrm{nat}(\beta,\sigma_1) \quad \mathrm{nat}(\sigma_1,\alpha)\,\mathrm{nat}(\beta,\sigma_1)\,\mathrm{nat}(\beta,\sigma_2)^2\big) = 0\ , \end{aligned}$$

as desired. $\diamond$

By taking transposes, it is immediately clear that the same holds true for the discrete differential: for each $\kappa > 0$, $B_\kappa B_{\kappa\ \ 1} = \quad$.

This fact, which we'll refer to as the *homology property*, is the starting point of discussing the properties of L_κ in **[65]**, including that $L_\kappa^\wedge$ and $L_\kappa^\vee$ are orthogonal, and yields the well-known *Hodge decomposition*:

$$\mathbb{R}^n = \mathrm{im}(B_{\kappa\ \ 1}) \oplus \mathrm{im}(B_\kappa) \oplus \ker(L_\kappa)\ . \tag{4.2}$$

One reason we emphasize this point is that for any choice of $a^\vee, a^\wedge \in \mathbb{R}$, the eigenvectors of $a^\vee L_\kappa^\vee + a^\wedge L_\kappa^\wedge$ are identical to those of L_κ, and those with nonzero eigenvalue are either purely an eigenvector of $L_\kappa^\vee$ or $L_\kappa^\wedge$; the corresponding eigenvalues for the linear combination are simply scaled by $a^\vee$ or $a^\wedge$ respectively. In **[14]**, $a^\vee, a^\wedge$ are restricted to be positive, and used to weigh the relative importance of the lower and upper κ-Laplacians in a spectral embedding using L_κ.

nother consequence of Lemma 4.1.1 is an explicit relationship between the off-diagonal elements of $L_\kappa^\wedge$ and $L_\kappa^\vee$. Given a simplicial complex C and $\sigma, \tau \in C_\kappa$, $\sigma \neq \tau$, let $[U]_{\sigma\tau} \coloneqq \mathbf{1}_{\sigma \simeq \tau}$ be the κ-hull matrix, and let $[A]_{\sigma\tau} \coloneqq \mathbf{1}_{\sigma \sim \tau}$ be the κ-adjacency matrix. Set $[A]_{\sigma\sigma} = [U]_{\sigma\sigma} = 0$. Let $[\overline{U}]_{\sigma\tau} \coloneqq 1 \quad [U]_{\sigma\tau}$ be the indicator for the *lack* of a κ-hull. Notice that $U = A \odot U$, where $\odot$ denotes the Hadamard (i.e., element-wise) product.

PROPOSITION 4.1.1. *Let $\sigma \neq \tau$ be κ-simplices. Then*

$$\left[L_\kappa^\vee\right]_{\sigma\tau} = \quad \mathrm{con}(\sigma,\tau)[A]_{\sigma\tau} \quad \textit{and} \quad \left[L_\kappa^\wedge\right]_{\sigma\tau} = \quad \mathrm{hcon}(\sigma,\tau)[U]_{\sigma\tau}\ ,$$

so that

$$[L_\kappa]_{\sigma\tau} = \quad \mathrm{con}(\sigma,\tau)[A]_{\sigma\tau}[\overline{U}]_{\sigma\tau}\ .$$

PROOF. On the one hand,

$$\left[L_\kappa^\vee\right]_{\sigma\tau} = \left[B_{\kappa\ \ 1} B_{\kappa\ \ 1}\right]_{\sigma\tau} = \sum_{\in C_{\kappa\ \ 1}} \mathrm{rel}(\sigma,\alpha)\,\mathrm{rel}(\tau,\alpha) = p_\sigma p_\tau \sum_{\in C_{\kappa\ \ 1}} \mathrm{nat}(\sigma,\alpha)\,\mathrm{nat}(\tau,\alpha)\ .$$

The term of the sum for a given α will vanish unless α is a face of both σ and τ; in other words, it vanishes unless $\sigma \sim \tau$, and only for the distinct term $\alpha^* = \mathrm{bd}(\sigma,\tau)$. Hence,

$$\left[B_{\kappa\ \ 1} B_{\kappa\ \ 1}\right]_{\sigma\tau} = \quad \mathrm{con}(\sigma,\tau)[A]_{\sigma\tau}\ .$$

On the other hand,

$$\Big[B_\kappa B_\kappa\Big]_{\sigma\tau} = \sum_{\beta\in C_{\kappa+1}} \mathrm{rel}(\beta,\sigma)\,\mathrm{rel}(\beta,\tau) = p_\sigma p_\tau \sum_{\beta\in C_{\kappa+1}} \mathrm{nat}(\beta,\sigma)\,\mathrm{nat}(\beta,\tau)\ .$$

The term of the sum for a given β will vanish unless β has both σ and τ as faces; in other words, it vanishes unless $\sigma \simeq \tau$, and only for the distinct term $\beta^* = \mathrm{hl}(\sigma,\tau)$. Hence,

$$\big[L_\kappa^\wedge\big]_{\sigma\tau} = \Big[B_\kappa B_\kappa\Big]_{\sigma\tau} = \quad \mathrm{hcon}(\sigma,\tau)U_{\sigma\tau} = \mathrm{con}(\sigma,\tau)[U]_{\sigma\tau}\ ,$$

and so finally,

$$[L_\kappa]_{\sigma\tau} = \quad \mathrm{con}(\sigma,\tau)\,([A]_{\sigma\tau} \quad [U]_{\sigma\tau}) = \quad \mathrm{con}(\sigma,\tau)[A]_{\sigma\tau}[\overline{U}]_{\sigma\tau}\ ,$$

as desired. $\diamond$

When constructing the graph Laplacian in the standard setting, the definition is usually given as $L_0 = D \quad A$ for an appropriate diagonal matrix D representing vertex degrees, and a (generally dense) matrix A representing adjacency. Now we can proceed in reverse, and infer the appropriate notion of adjacency between two κ-simplices from the κ-Laplacian.

First, we compute the lower degrees and upper degrees, respectively, which together give the diagonal degree matrix:

$$\big[L_\kappa^\vee\big]_{\sigma\sigma} = \sum_{\in C_\kappa\ \ 1} \mathrm{rel}(\sigma,\alpha)^2 = \sum_{\in C_\kappa\ \ 1} \mathbf{1}_{\{\quad \text{is} \quad \text{f ce of } \sigma\}} = \kappa+1\ ,$$

and

$$\big[L_\kappa^\wedge\big]_{\sigma\sigma} = \sum_{\beta\in C_{\kappa+1}} \mathrm{rel}(\beta,\sigma)^2 = \sum_{\beta\in C_{\kappa+1}} \mathbf{1}_{\{\sigma \text{ is} \quad \text{f ce of } \beta\}} = \frac{1}{\kappa+1}\,[U\mathbf{1}]_\sigma\ .$$

Denote the corresponding diagonal matrix $D_\kappa \coloneqq \mathrm{diag}((\kappa+1)\mathbf{1} + \frac{1}{\kappa+1}U\mathbf{1})$. Next, consider the off-diagonal elements of L_κ. Notice that the non-negative component $A\odot\overline{U}$ is nonzero only for simplices which are adjacent, but not strongly-adjacent, so precisely when they are κ-adjacent. Now, defining the signed κ-adjacency matrix S_κ via

$$[S_\kappa]_{\sigma\tau} = \begin{cases} \mathrm{con}(\sigma,\tau) & \sigma \underset{\kappa}{\sim} \tau \\ 0 & \text{otherwise} \end{cases}\ ,$$

we conclude that $L_\kappa = D_\kappa \quad S_\kappa$. Hence, *the combinatorial κ-Laplacian represents signed adjacency between κ-adjacent simplices, via their consistency.* In particular, this means that given a simplicial complex C, L_κ can be computed from just natural orientations, which depend only on the simplex vertices, and the orientations of each κ-simplex in C_κ.

Naively, constructing the boundary matrices $B_{\kappa\ 1}, B_\kappa$ then requires superfluous sign information – the orientation of each member of both $C_{\kappa\ 1}$ and $C_{\kappa+1}$. This situation exactly mirrors that of the graph Laplacian L_0. In order to construct L_0 for an undirected graph from boundary matrices, one must assign an arbitrary direction to each edge, and the resulting Laplacian is independent of that choice of directions. In the language of κ-simplices, one must assign to each 1-hull an arbitrary orientation consistent with the natural orientation, i.e., so that each resulting oriented 1-simplex has one head and one tail, and the resulting κ-Laplacian is then independent of the choice of orientations.

This is clear from the relevant linear algebra as well; if $\boldsymbol{s} \in (\pm 1)^{|V(G)|}$, and $\Sigma = \mathrm{diag}(\boldsymbol{s})$, then $\Sigma\Sigma \ = \Sigma^2 = I$, and the result of changing the orientations in G such that edge e_i is flipped when $\boldsymbol{s}_i < 0$, is

$$\tilde{B_0} = B_0\Sigma \ , \text{ such that } \tilde{L_0} = \tilde{B_0}\tilde{B_0} \ = B_0\Sigma^2 B_0 \ = B_0 B_0 \ = L_0 \ .$$

Including these arbitrary orientations obscures the fact that the Laplacian doesn't depend on them.

4.1.1. Weighted and normalized κ-Laplacians.

We will motivate the introduction of a *weighted* simplicial complex, by first considering the symmetrically normalized graph Laplacian

$$L_0^{\mathrm{sym}} \coloneqq {D_0}^{1/2} B_0 D_1 B_0 \, {D_0}^{1/2} = \left({D_0}^{1/2} B_0 D_1^{1/2}\right)\left({D_0}^{1/2} B_0 D_1^{1/2}\right) \quad ,$$

where D_0, D_1 are diagonal matrices, whose diagonal contains the weights of vertices (C_0) and edges (C_1) respectively. In this discussion only, when σ is a κ-simplex, we will abuse notation and use d_σ to refer to the appropriate entry in D_κ, and use $\sigma \sim \alpha$ to refer to α being a face of σ. For each vertex-edge pair (v, e) we can associate the weight $r_{ev} \coloneqq \sqrt{d_e/d_v}$, and then when $e = (u, v)$, the weight of the vertex pair (u, v) is the product $r_{ev} r_{eu}$. The diagonal of L_0^{sym} has the form

$$[L_0^{\mathrm{sym}}]_{uu} = \sum_{e \sim u} r_{eu}^2 \ ,$$

and a standard consequence of normalizing the Laplacian is to take this sum to be 1 for each u, which in turn forces $d_u = \sum_{e \sim u} d_e$; i.e., the standard assumption that vertex weights are constrained to be the sum of all adjacent edge weights. We can write this constraint as $D_0 = \mathrm{diag}(|B_0| D_1 \mathbf{1})$, or equivalently $D_0 \mathbf{1} = |B_0| D_1 \mathbf{1}$, with the absolute value taken element-wise.

Now, letting D_κ generally refer to a diagonal matrix containing κ-simplex weights, we proceed as in **[14]** and define the *symmetrically normalized κ-Laplacian* as

$$L_\kappa^{\mathrm{sym}} \coloneqq \quad _\kappa \ _1 \quad \kappa \ _1 + \quad \kappa \quad _\kappa \ ,$$

where $\quad _\kappa \coloneqq D_\kappa \ ^{1/2} B_\kappa D_{\kappa+1}^{1/2}$. The weight for a pair of simplices is now composed of both an upper (hull) and lower (boundary) weight, and from Proposition 4.1.1, we know that the total weight is the *difference* of these, even while their contribution to the diagonal degree term is the *sum*.

Notice that for the upper normalized Laplacian, the condition $\mathrm{diag}(\quad _\kappa \quad _\kappa) = \mathbf{1}$ is, just as in the $\kappa = 0$ case, equivalent to the constraint

$$D_\kappa \mathbf{1} = |B_\kappa| D_{\kappa+1} \mathbf{1} \ , \tag{4.3}$$

so that the simplex weights are just degrees of the hull weights. For the lower normalized Laplacian, in **[14]**, they constrain the $(\kappa \quad 1)$-simplex boundary weights in the analogous way, with $D_{\kappa \ _1} \mathbf{1} = |B_{\kappa \ _1}| D_\kappa \mathbf{1}$, so that boundary weights are themselves degrees, summing the simplex degrees in turn.

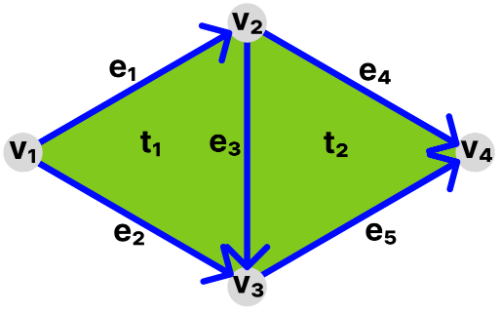
different principled choice is to observe that the lower Laplacian flips the roles of boundary and hull, so that when we apply the condition $\mathrm{diag}(\quad _\kappa \ _1 \quad \kappa \ _1) = \mathbf{1}$, we find

$$[\quad _\kappa \ _1 \quad \kappa \ _1]_{\sigma\sigma} = d_\sigma \sum_{\sim \sigma} d \ ^1 = 1 \ ,$$

so that $d_\sigma = \left(\sum \ _{\sim \sigma} d \ ^1 \right) \ ^1 = h(\{d \quad | \ \alpha \sim \sigma\})$, the *harmonic sum* over the simplex degrees, instead of directly taking their sum. Written in matrix form, this creates the constraint

$$D_\kappa \ ^1 \mathbf{1} = |B_{\kappa \ _1}| \ D_\kappa \ ^1 {}_1 \mathbf{1} \ . \tag{4.4}$$

Now, these two constraints on $D_\kappa \mathbf{1}$ suggest two ways to proceed. First, we may assume that D_κ is fixed by the upper Laplacian constraint, and then solve the lower Laplacian constraint, say



$$S_1^{\mathrm{wt}} = \frac{1}{4}\begin{bmatrix} 0 & 2 & 1 & 1 & 0 \\ 2 & 0 & 2 & 0 & 1 \\ 2 & 2 & 0 & 2 & 2 \\ 1 & 0 & 2 & 0 & 2 \\ 0 & 1 & 2 & 2 & 0 \end{bmatrix}$$

FIGURE 4.3. The complex from Figure 2.6 on the left, with natural orientation displayed as directed edges, together with its weighted, unnormalized signed adjacency matrix S_1^{wt}, with $D_2 = I$. Notice that weights differ depending on consistency and presence or lack of hull, and that the presence of a hull can switch the expected sign.

in a least-squares sense, to fix $D_{\kappa\ 1}$. This leaves the hull weights $D_{\kappa+1}$ free to vary, as input data to constructing L_κ^{sym}. lternatively, we can assume that D_κ is fixed by the *lower* Laplacian constraint, and then solve the upper Laplacian constraint to fix $D_{\kappa+1}$. This provides a principled way to construct the normalized, weighted κ-Laplacian when the weight data is supplied on the boundaries, instead of on the hulls.

From L_κ^{sym} we may define the usual weighted un-normalized κ-Laplacian L_κ^{wt}, and the random-walk normalized L_κ^{rw}, whose eigenvectors will be the basis for our bipartitioning:

$$L_\kappa^{\mathrm{wt}} \coloneqq D_\kappa^{1/2} L_\kappa^{\mathrm{sym}} D_\kappa^{1/2} \quad \text{and} \quad L_\kappa^{\mathrm{rw}} \coloneqq D_{\kappa\ }^{1} L_\kappa^{\mathrm{wt}} \ .$$

One reason we didn't introduce L_κ^{wt} first is that it lacks many of the nice properties of L_κ. First, from expanding the definition

$$L_\kappa^{\mathrm{wt}} \coloneqq D_\kappa^{1/2} L_\kappa^{\mathrm{sym}} D_\kappa^{1/2} = D_\kappa B_{\kappa\ 1} D_{\kappa\ 1}^{\ 1} B_{\kappa\ 1} D_\kappa + B_\kappa D_{\kappa+1} B_\kappa \ ,$$

it is clear that we cannot both recover the combinatorial L_κ, and satisfy either of the degree constraints (4.3), (4.4); we would need to have that $D_{k+1} = I$, and that for each pair of adjacent simplices σ, τ with boundary α, that $d_\sigma d_\tau = d$. lso, L_κ^{wt} is the only κ-Laplacian variant discussed so far which does *not* benefit from the homology property (4.1), since $D_{\kappa\ 1}^{\ 1/2} B_{\kappa\ 1} D_\kappa B_\kappa D_{\kappa+1}^{1/2} \neq$. This means that the lower and upper un-normalized, weighted κ-Laplacians are not orthogonal, linear combinations of them may not have the same eigenvectors, and the L_κ^{wt}-harmonic functions,

those f satisfying $L_\kappa^{\mathrm{wt}} f = \quad$, are not simply characterized by the intersection of their null spaces. For these reasons, we do not use L_κ^{wt} for analysis. nalogously, we also do not explore normalized combinatorial k-Laplacians of the form $DL_\kappa E$ for some diagonal matrices D, E, as such a normalization will not generally have the homology property either.

This exploration shows that both choice of weight and degree conventions, and choice of normalization have more degrees of freedom for simplicial complexes than for graphs, and that the kind to use may depend on the problem, whether weights are relevant, and how they should be interpreted. For example, if a simplicial complex is equipped with *both* ($\kappa \quad 1$)-boundary weights and ($\kappa+1$)-hull weights without a clear relationship between them, then we define the *decoupled* weighting of L_κ as

$$L_\kappa^{\mathrm{dc}} = B_{\kappa \ 1} D_{\kappa \ 1} B_{\kappa \ 1} + B_\kappa D_{\kappa+1} B_{\kappa \ },$$

so that the familiar "degree diagonal minus weights" formulation applies to both the upper and lower κ-Laplacian separately. The decoupled κ-Laplacian satisfies the homology property, and while one can construct a normalized version by normalizing the lower and upper κ-Laplacians separately, there is then no simple relationship with the normalized and un-normalized version, and further, the normalized version loses the homology property.

While in the combinatorial case, L_κ vanishes for pairs $\sigma \simeq \tau$, each of the weighted Laplacians $L_\kappa^{\mathrm{wt}}, L_\kappa^{\mathrm{rw}}, L_\kappa^{\mathrm{sym}}, L_\kappa^{\mathrm{dc}}$, and any linear combination of lower and upper Laplacian variant with coefficients $a^\vee, a^\wedge$ unequal may be nonzero whenever $\sigma \sim \tau$. Finally, we define the weighted analogues of the signed adjacency matrices, $S_\kappa^{\mathrm{wt}}, S_\kappa^{\mathrm{sym}}, S_\kappa^{\mathrm{rw}}, S_\kappa^{\mathrm{dc}}$, as the negatives of the off-diagonal parts of their respective Laplacians.

4.2. κ-Fiedler Vector

Let C be a simplicial complex, such that $G = (C_0, C_1)$ is a connected graph. For a given κ, let $\boldsymbol{p}$ be a vector of orientations over C_κ, with each $[\boldsymbol{p}]_\sigma \in \{\pm 1\}$, and let $\Sigma_p = \mathrm{diag}(\boldsymbol{p})$. Let L_κ^{wt}, $\tilde{L}_\kappa^{\mathrm{wt}}$ denote the weighted κ-Laplacian of C_κ with natural orientations, and with orientations given by $\boldsymbol{p}$, respectively. Let $\lambda_0 \leq \cdots \leq \lambda_{n \ 1}$ be the eigenvalues of L_κ^{wt} and $\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{n \ 1}$ be the corresponding eigenvectors where $n = |C_0|$. Then, let $(\tilde{\lambda}_i, \tilde{\boldsymbol{\phi}}_i)$ be the eigenpairs for $\tilde{L}_\kappa^{\mathrm{wt}}$. Because $\tilde{L}_\kappa^{\mathrm{wt}} = \Sigma_p L_\kappa^{\mathrm{wt}} \Sigma_p$, then $\tilde{\lambda}_i = \lambda_i$ and $\tilde{\boldsymbol{\phi}}_i = \Sigma_p \boldsymbol{\phi}_i$ for $0 \leq i < n$.

For $\kappa = 0$, with the vertices of G in natural orientation, we have that $\lambda_0 = 0$, $\lambda_1 > 0$, $\boldsymbol{\phi}_0 = \mathbf{1}$ and in particular is non-oscillatory, and that $\boldsymbol{\phi}_1$ acts as a single global oscillation, appropriate to bipartition the vertices of G with. Considering $\tilde{L}_0^{\text{wt}}$ for nontrivial $\boldsymbol{p} \notin \{\pm\mathbf{1}\}$, $\tilde{\boldsymbol{\phi}}_0$ is oscillatory, and $\tilde{\boldsymbol{\phi}}_1$ is no longer appropriate for clustering; this is one reason that oriented 0-simplices are always considered to be in natural orientation.

For $\kappa > 0$ however, it is no longer true that $\boldsymbol{\phi}_0$ will be non-oscillatory. Let $\boldsymbol{p}^*$ be a vector of orientations such that where $[\boldsymbol{\phi}_0]_\sigma \neq 0$, $[\boldsymbol{p}^*]_\sigma = \text{sgn}([\boldsymbol{\phi}_0]_\sigma)$. Then the corresponding $\tilde{\boldsymbol{\phi}}_0$ *is* non-oscillatory, and acts as a DC component. This motivates taking $\text{sgn}(\boldsymbol{\phi}_0) \odot \boldsymbol{\phi}_1$ (element-wise) as the Fiedler vector of L_κ^{wt}, with which to bipartition C_κ.

In order to build intuition for this choice of Fiedler vector, we consider some simple simplicial complexes, the κ-paths. The 0-path graph P_n is a graph on n 0-simplices, with $n \quad 1$ hulls connecting each in order, and its Laplacian eigenvectors, as discussed in depth in Section 3.2, are a sequence of uniform oscillations (i.e., DCT basis vectors), increasing in frequency with increasing eigenvalue, whose signs therefore yield clear partitions of the path. Notice that the 0-path may also be considered a 1-complex, whose 1-simplices are weakly adjacent along a sequence of vertices. We will call this type of κ-path, connecting n κ-simplices along some mutual, unrepeated boundaries, and containing no $(\kappa+1)$-hulls, a *lower simplex path*, denoted $P_{n,\kappa}^{\downarrow}$. The complex $P_{n,\kappa}^{\downarrow}$ consists of $n+\kappa$ vertices, and the n simplices are connected by a total of $1+n\kappa$ boundary faces.

Then let $P_{n,\kappa}$ refer to the *simplex path*, which consists of n connected $(\kappa+1)$-hulls, each adjacent to a distinct pair of κ-simplices. $P_{n,\kappa+1}^{\downarrow}$ has the same graph structure as $P_{n,\kappa}$; for the former, we are interested in the $(\kappa+1)$-simplices, and in the latter, the κ-simplices. For $\kappa = 0$, our notation differs slightly from the graph setting in that $P_{n,0} = P_{n+1}$. The complex $P_{n,\kappa}$ consists of $n+\kappa+1$ vertices, joining $1+n(\kappa+1)$ simplices. Successive sets of $\kappa+1$ simplices (each in C_κ) form the faces of each of the n hulls (each in $C_{\kappa+1}$) of $P_{n,\kappa}$, and they are connected by a total of $\kappa+1+n\binom{\kappa+1}{2}$ boundary faces (each in $C_{\kappa \ 1}$).

Figure 4.4 displays the bottom eigenvectors of L_κ for $P_{50,2}^{\downarrow}$, and Figure 4.5 displays the same for $P_{50,1}$. In both cases, the similarity with the $\kappa = 0$ path case are clear, and $\boldsymbol{\phi}_1$ behaves appropriately as a Fiedler vector. The latter example especially illustrates that eigenvalue alone is a poor way to organize the eigenvectors, because of their inherent higher-dimensional organization; in this case

because the graph underlying the simplicial complex is a variant of a two-dimensional grid. Proper organization of the eigenvectors and the pitfalls of using solely eigenvalue are explored in [**17**, **75**].

We also note that L_κ and either L_κ^{sym} or L_κ^{wt} may yield dramatically different results when simplices in the κ-region have no adjacent hulls, and sometimes the latter matrix will have undefined values, when a (κ 1)-boundary face is adjacent only to κ-simplices with no hulls. The lower simplex path is an extreme example, with no hulls at all; equivalently, in the combinatorial case for these complexes we analyze only the lower Laplacian. Practically, when computing weighted Laplacians in our `k_laplacian` code, we include the option `weakadjweight` to add a uniform weight ϵ for every weakly-adjacent pair, which prevents the appearance of zero degrees on either κ-simplices or (κ 1)-boundary faces. This modifies the simplex degree constraint to be

$$D_\kappa \mathbf{1} = \left(|B_\kappa| D_{\kappa+1} + \epsilon |B_{\kappa \ 1}| \right) \mathbf{1} \ . \tag{4.5}$$

In the $\kappa = 0$ case, this would be equivalent to adding a constant weight to every edge, with the corresponding increase appearing in all vertex degrees.

Now, what if G is not connected? In the $\kappa = 0$ case, the behavior of eigenvectors with eigenvalue $\lambda_0 = 0$ is well understood, and in general, the Fiedler vector is not $\boldsymbol{\phi}_1$, but $\boldsymbol{\phi}_{\mathfrak{n}+1}$, where $\mathfrak{n}$ is the nullity of L_0. For general κ, the nullity of L_κ for a given simplicial complex is known to be the κ^{th} *etti number* $\mathfrak{b}_\kappa$, which counts the number of "κ-dimensional holes" in the complex [**33**]. This is an explicit generalization, as $\mathfrak{b}_0 = \mathfrak{n}$. While we don't dive into the topological properties of L_κ in this dissertation, it is enough to notice that harmonic functions do not generally have global support, and so make for poor DC components. In the simplex path examples, the relevant complexes all have strictly positive definite Laplacians, and so no eigenvectors are harmonic. In Figure 4.6, we demonstrate how blindly taking $\text{sgn}(\boldsymbol{\phi}_0) \odot \boldsymbol{\phi}_1$ fails to produce a meaningful partition for a less structured complex, but that taking the first non-harmonic eigenvector as the DC component yields clear oscillations. Hence, *for general complexes, we will take* $\text{sgn}(\boldsymbol{\phi}_{\mathfrak{b}_\kappa+1}) \odot \boldsymbol{\phi}_{\mathfrak{b}_\kappa+2}$ *as the Fiedler vector.*

We will aim to bipartition κ-regions by following a standard strategy in spectral clustering, of minimizing a relaxation of a combinatorial cut function over possible partitions. Just as a graph cut is typically defined as the volume of edge weight which crosses a partition of the vertices, we

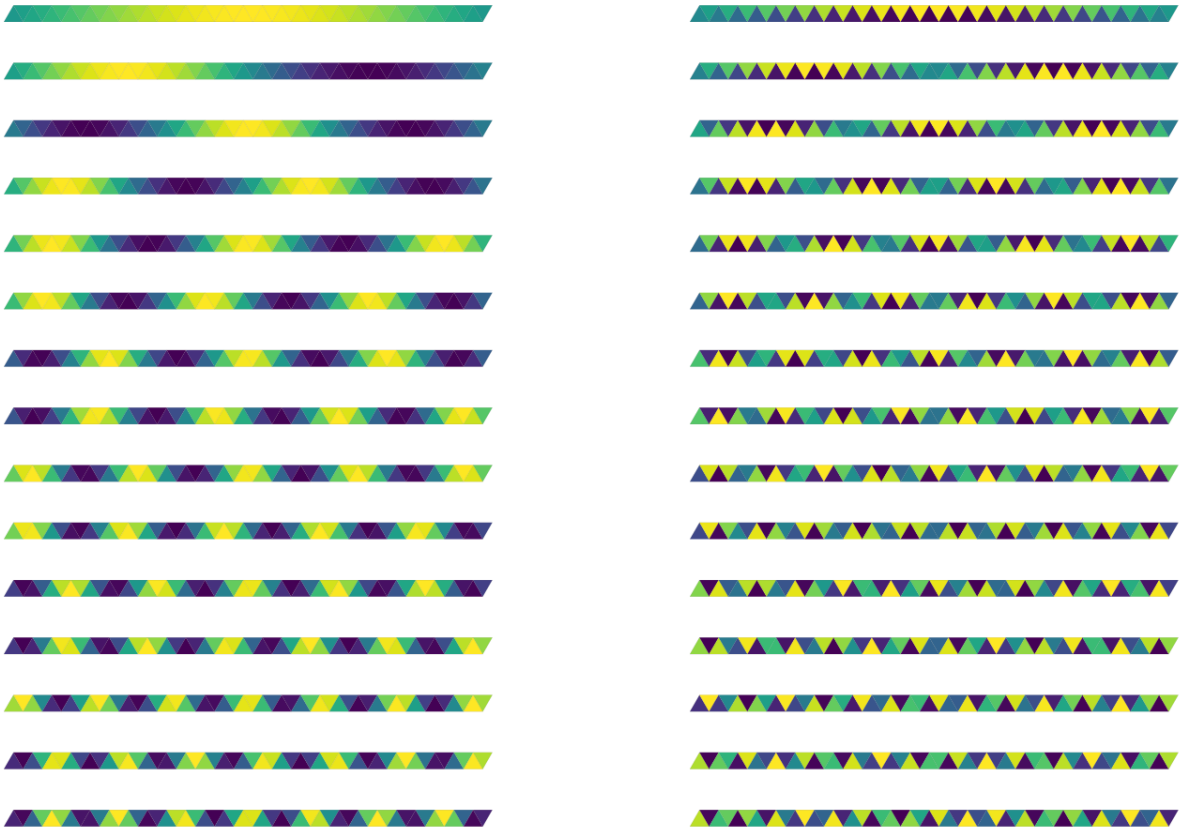


FIGURE 4.4. visualization of the first fifteen eigenvectors of $L_\kappa = L_2$ for $P^{\downarrow}_{50,2}$. On the left, orientations have been chosen for the triangles such that ϕ_0 has consistent sign, while on the right, the triangles have been left in natural orientation. Yellow indicates positive values, and purple indicates negative values. Notice how ϕ_1 for the re-oriented complex behaves precisely as one expects here for a Fiedler vector.

can define the positive and negative *signed consistency cut* of C_κ into subregions A, B as

$$\mathrm{Ccut}_\kappa{}^{\pm}(A,B) \coloneqq \sum_{\substack{\sigma\in A,\tau\in B\\ \sigma\sim\tau}} [S^{\pm}_\kappa]_{\sigma\tau} \ ,$$

where $[S^{\pm}_\kappa]_{\sigma\tau} = \max(0, \pm[S^{\mathrm{wt}}_\kappa]_{\sigma\tau})$ are indicator functions for consistent/inconsistent pairs, respectively[1]. Because of the signs introduced by consistency, we consider S^{wt}_κ as the signed, weighted adjacency matrix for a signed graph over C_κ, and so can utilize the framework of *signed* Laplacians **[62]**.

[1]This argument also goes through identically with S^{wt}_κ replaced by the combinatorial S_κ.

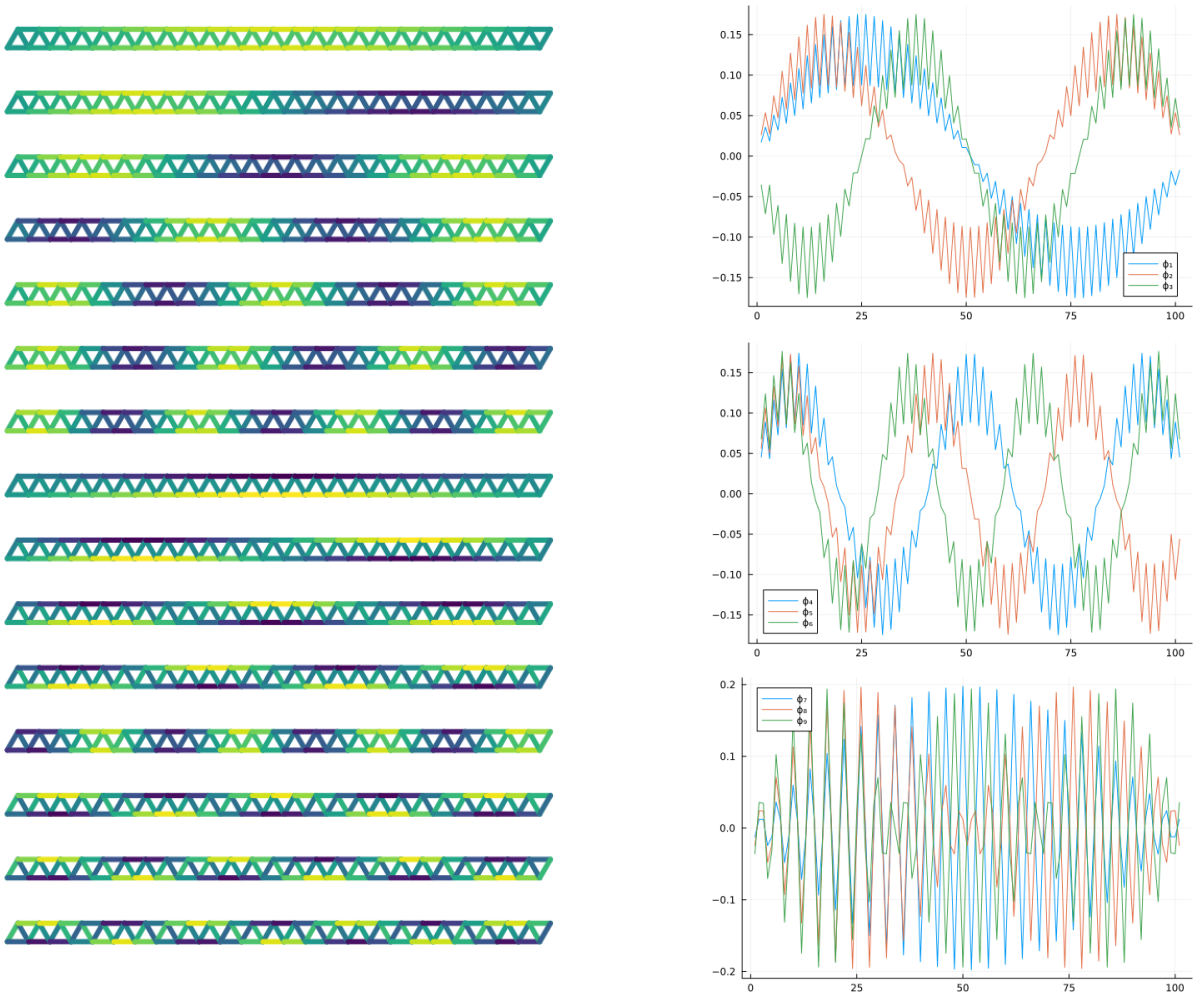


FIGURE 4.5. On the left, a visualization of the first fifteen eigenvectors of $L_\kappa = L_1$ for $P_{50,1}$. In this case, the natural orientations are already such that ϕ_0 has constant sign. There are two types of oscillations, analogous to the L_0-eigenvectors of a ladder graph, or a narrow grid graph. Eigenvectors 0 6 oscillate along the length of the path in the expected way, then 7 10 include a vertical oscillation which assigns opposite sign to the two outer 0-paths. Higher eigenvectors continue adding either vertical or horizontal oscillations. On the right, different oscillation types are illustrated, for eigenvectors 1 3, 4 6, and 7 9. The sub-oscillations are a result of the edge indexing, because there are three types of edges interspersed with each other: those on one of the two outer 0-paths, and those on the inner 0-path.

Define the *consistency volume* $\mathrm{Cvol}_\kappa{}^{\pm}(A) \coloneqq \mathrm{Ccut}_\kappa{}^{\pm}(A, A)$, $\mathrm{Cvol}_\kappa(A) \coloneqq \mathrm{Cvol}_\kappa{}^{+}(A) + \mathrm{Cvol}_\kappa \quad (A)$,

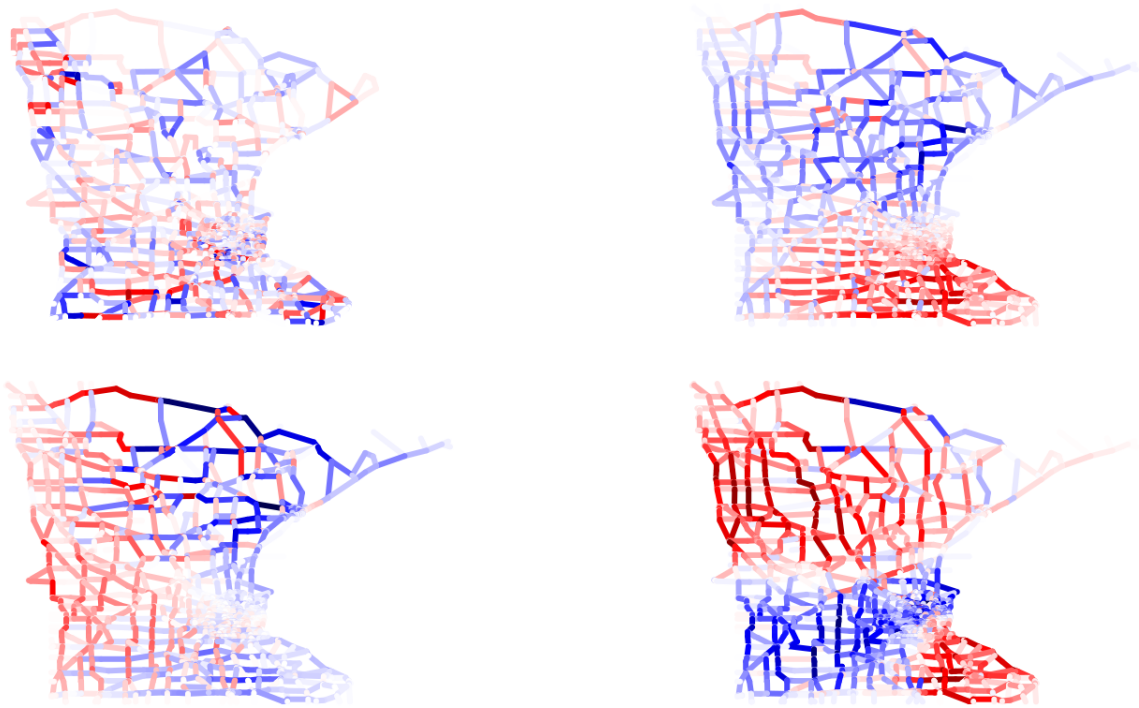


FIGURE 4.6. Various edge eigenvectors of L_1, for the clique complex constructed from the Minnesota road network. Here $\mathfrak{b}_1 = 610$, so the 0-eigenspace is highly degenerate, as the road network contains many cycles, but few triangles. More red indicates more positive, and more blue indicates more negative values. The top left is the standard Fiedler vector $\boldsymbol{\phi}_1$, in an orientation that makes $\boldsymbol{\phi}_0$ non-negative. This harmonic vector yields no clear partition or structure of the graph. The other three are in an orientation that makes $\boldsymbol{\phi}_{611}$ non-negative, and from left to right, top to bottom, are eigenvectors $\boldsymbol{\phi}_{613}, \boldsymbol{\phi}_{614}, \boldsymbol{\phi}_{615}$. The progression clearly demonstrates meaningful partitions on the 1-region, with increasing oscillation.

and the *signed κ-cut*

$$\kappa\mathrm{Cut}(A, B) \coloneqq 2\,\mathrm{Ccut}_\kappa{}^+(A, B) + \mathrm{Cvol}_\kappa\ \ (A) + \mathrm{Cvol}_\kappa\ \ (B)\ .$$

In the $\kappa = 0$ case, with all vertices in natural orientation, S_0^{wt} is just the usual adjacency matrix, and so $S_0 \ = \ $; hence $\kappa\mathrm{Cut} = 2\,\mathrm{Ccut}_\kappa$, yielding the traditional cut objective. For $\kappa > 0$, κCut increases with the number of consistent pairs of κ-adjacent simplices across the partition, and with the number of inconsistent pairs within each κ-region. Equivalently, minimizing κCut requires maximizing consistent pairs within each κ-region, and maximizing inconsistent pairs across the partition.

Let $\overline{L}_\kappa$ be the signed Laplacian with signed adjacency S_κ^{wt}. Let A be a κ-region, define $\boldsymbol{r}_A \coloneqq \mathbf{1}_A \quad \mathbf{1}_{C_\kappa \setminus A}$, and $R_A(L) \coloneqq \boldsymbol{r}_A L \boldsymbol{r}_A$. Then because $\overline{L}_\kappa$ differs from L_κ^{wt} only on the diagonal, $R_A(\overline{L}_\kappa)$ differs from $R_A(L_\kappa^{\mathrm{wt}})$ by a constant independent of A. From **[62]**, we know that $R_A(\overline{L}_\kappa) \propto \kappa\mathrm{Cut}(A, C_\kappa \setminus A)$. Hence, $\min_{A \subset C_\kappa} R_A(L_\kappa^{\mathrm{wt}}) = \min_{A \subset C_\kappa} \kappa\mathrm{Cut}(A, C_\kappa \setminus A)$, and we obtain $\boldsymbol{\phi}_0$ as a relaxed solution to signed κ-cut minimization.

Now, notice that if the orientations of C_κ were changed according to some $\boldsymbol{p}$, this would be equivalent to a different choice of A; namely, if $[\boldsymbol{p}]_\sigma = \quad 1$, then σ moves to the other side of the partition, either into or out of A. s all orientations are available to us, this includes one for which $\tilde{\boldsymbol{\phi}}_0$ is non-oscillatory, so that its sign does not partition C_κ. We then instead take $\tilde{\boldsymbol{\phi}}_1$ as our relaxed solution, which we may compute via $\mathrm{sgn}(\boldsymbol{\phi}_0) \cdot \boldsymbol{\phi}_1$.

Finally, we need to avoid circumstances where some or many entries of the Fiedler vector vanish, because those entries provide no partitioning information (or in practice, random partitioning, from noise in the eigenvector computation). In the $\kappa = 0$ case, this is handled by skipping the first $\mathfrak{n}$ eigenvectors, first to ensure orthogonality to $\mathbf{1}$, but also because the eigenspace of $\lambda_0 = 0$ is spanned by the indicator functions of connected components, which then vanish for the rest of the vertices in the graph.

In **[23**, Ch. 8], it is shown that the eigenspace for L_κ of $\lambda = 0$ can be spanned by *harmonic components*, which vanish except near a hole in the graph caused by a minimal cycle that is not a triangle. Due to their sparsity, even if these eigenvectors optimize κCut, they do so trivially, so we discard them, again arriving at the general relaxed solution $\mathrm{sgn}(\boldsymbol{\phi}_{\mathfrak{b}_\kappa+1}) \odot \boldsymbol{\phi}_{\mathfrak{b}_\kappa+2}$.

n improved cut objective is the *signed Ratio Cut*, which encourages more balanced partitions:

$$\mathrm{SignedRatioCut}(A) \coloneqq \left(\frac{1}{|A|} + \frac{1}{|C_\kappa \setminus A|}\right) \kappa\mathrm{Cut}(A, C_\kappa \setminus A) \ .$$

From **[62]**, we know that with $\boldsymbol{r}_A$ above scaled by a factor of $c_A \coloneqq \sqrt{|A|/|C_\kappa \setminus A|}$, the analogous result holds, that the eigenvectors of $\overline{L}_\kappa$ yield a relaxed solution to $\min_{A \subset C_\kappa} \mathrm{SignedRatioCut}(A)$. However, the new dependence on A means the resulting objective is slightly different for L_κ, so the relaxation is only approximate.

FIGURE 4.7. One possible hierarchical bipartitioning of a simple 2-complex, from $j = 0$ with no partition on the left, to $j = 5$ on the right, where each of the 21 triangles form their own subregion. Colors indicate distinct subregions.

Finally, the *signed Normalized Cut* balances the partitions by degree rather than simplex count:

$$\text{SignedNormalizedCut}(A) \coloneqq \left(\frac{1}{\text{Cvol}_\kappa(A)} + \frac{1}{\text{Cvol}_\kappa(C_\kappa \setminus A)}\right) \kappa\text{Cut}(A, C_\kappa \setminus A).$$

Here, the eigenvectors of $\text{diag}(\overline{L}_\kappa)^{\dagger 1}\overline{L}_\kappa$ yield a relaxed solution to $\min_{A \subset C_\kappa} \text{SignedNormalizedCut}(A)$, and an approximate relaxed solution is given by the appropriate eigenvectors of L_κ^{rw}, again $\text{sgn}(\boldsymbol{\phi}_{\mathfrak{b}_\kappa+1}) \odot \boldsymbol{\phi}_{\mathfrak{b}_\kappa+2}$. In our numerical experiments, we use the random-walk κ-Laplacian for bipartitioning simplicial complexes unless we say otherwise.

4.3. κ-Haar Basis

This section borrows from and builds on the corresponding section in our preprint [**79**].

With the κ-Fiedler vector in hand, we can proceed to recursively bipartition C_κ to form a hierarchical bipartition tree. This bipartitioning operation ideally splits each κ-subregion into two smaller κ-subregions that are roughly equal in size while keeping tightly-connected κ-simplices grouped together. More specifically, let C_k^j denote the kth κ-subregion on level j of the binary partition tree of C_κ and $n_k^j \coloneqq \left|C_k^j\right|$, where $j, k \in \mathbb{Z}_{\geq 0}$. Note $C_0^0 = C_\kappa$, $n_0^0 = n$, i.e., level $j = 0$ represents the root vertex of this tree. Then the two children of C_k^j in the tree, $C_{k'}^{j+1}$ and $C_{k'+1}^{j+1}$, are obtained through partitioning C_k^j using the Fiedler vector of $L_\kappa^{\text{rw}}\left(C_k^j\right)$. This partitioning is recursively performed until each κ-subregion corresponding to the leaf contains only a simplex singleton. Note that $k' = 2k$ if the resulting binary partition tree is a perfect binary tree. We note that as before, even other (non-spectral) partitioning methods can be used to form the binary partition tree, but here, we stick with the spectral clustering using the Fielder vectors. Figure 4.7 demonstrates such a hierarchical bipartition tree for a simple 2-complex consisting of triangles.

The usual definitions of Haar wavelets do not generalize to non-homogeneous domains due to the lack of appropriate translation operators and dilation operators [**85**]. Instead, several methods

have been proposed to generate similar bases, and overcomplete dictionaries, in order to build them on more abstract domains such as graphs and discretized manifolds [**47**, **80**, **86**]. Here, we describe a method to compute similar, piecewise-constant locally supported bases for κ-simplex valued functional spaces, which we call the (orthonormal) *κ-Haar bases*.

Rather than basing our construction on some kind of translation or transportation schemes, we instead employ the hierarchical bipartition, as we discussed in Section 2.3, to divide the domain, i.e., the κ-simplices C_κ of a given simplicial complex C into appropriate locally-supported κ-regions. For each κ-region in the bipartition tree, if that region has two children in the tree, then we create a vector that is positive on one child, negative on the other, and zero elsewhere. To avoid sign ambiguity, we dictate that the positive portion is on the region whose region index is smaller among these two.

Several remarks on this basis are in order. First, since the division is not symmetrically dyadic, we need to compute the scaling factor for each region separately. For each given basis vector $\boldsymbol{\xi}$ except the scaling vector, we break it into positive and negative parts $\boldsymbol{\xi}^+$ and $\boldsymbol{\xi}^-$ and ensure that $\sum_i([\boldsymbol{\xi}^+]_i + [\boldsymbol{\xi}^-]_i) = 0$ and $\|\boldsymbol{\xi}\| = 1$. If the members of the κ-region are weighted, then this sum and norm can be computed with respect to those weights. Finally, we note that different hierarchical bipartition schemes may arise from the different weighting of the κ-Laplacian, which will correspond to bases with different supports. Figure 4.8 demonstrates the 2-Haar basis on the simple 2-complex used in Figure 4.7, which has a hole in the center.

FIGURE 4.8. The 2-Haar basis vectors on the same simple 2-complex shown in Figure 4.7. The yellow, dark green, violet regions in each vector indicate its positive, zero, and negative components.

CH PTER 5

Multiscale -Region Transforms

Now, we introduce the two overcomplete dictionaries we will use for analyzing real-valued functions defined on κ-simplices in a given simplicial complex: the *κ-Hierarchical Laplacian Eigen Transform* (κ-HGLET), based on the *Hierarchical Graph Laplacian Eigen Transform* (HGLET) [**48**] and the *κ-Generalized Haar-Walsh Transform* (κ-GHWT), based on the *Generalized Haar-Walsh Transform* (GHWT) [**47**] for graph signals [1]. This material was also developed in the preprint [**79**].

5.1. Hierarchical κ-Laplacian Eigen Transform (κ-HGLET)

The first overcomplete transform we describe can be viewed as a generalization of the Hierarchical Block Discrete Cosine Transform (HBDCT). The classical HBDCT is generated by creating a hierarchical bipartition of the signal domain and computing the DCT of the local signal supported on each subdomain. We note that a specific version of the HBDCT (i.e., a homogeneous split of an input image into a set of blocks of size 8×8 pixels) has been used in the JPEG image compression standard [**71**]. This process was generalized to the graph case in [**48**], with the *Hierarchical Graph Laplacian Eigen Transform* (HGLET), from which we base our algorithm and notation. The basis given by the set $\{\boldsymbol{\phi}_{k,l}^{j}\}$ where j denotes the level of the partition (with $j = 0$ being the root), k indicates the partition within the level, and l indexes the elements within each partition in increasing frequency.

To compute the transform, we first compute the complete set of eigenvectors $\{\boldsymbol{\phi}_{0,l}^{0}\}_{l=1:n}$ of the κ-Laplacian for the entire κ-region C_κ of a given simplicial complex and order them as usual by non-decreasing eigenvalue. We then partition C_κ into two disjoint κ-regions C_0^1 and C_1^1 as described in Section 4.2. Next, we compute the complete set of eigenvectors of the κ-Laplacian on C_0^1 and C_1^1. We again order each set by non-decreasing eigenvalue and label these $\{\boldsymbol{\phi}_{0,l}^{1}\}_{l=1:n_0^1}$ and $\{\boldsymbol{\phi}_{1,l}^{1}\}_{l=1:n_1^1}$.

[1] We use -HGLET instead of, e.g., -HLET in order to maintain naming consistency with the corresponding $= 0$ graph transformations.

FIGURE 5.1. 2-HGLET dictionary on the 2-complex shown in Figure 4.7. Here, the color scale is consistent across each row (which corresponds to the level) to better visualize the smoothness of the elements

Note that $n_0^1 + n_1^1 = n_0^0 = n$, and that all of the elements in $\{\boldsymbol{\phi}_{0,l}^1\}$ are orthogonal to those in $\{\boldsymbol{\phi}_{1,l}^1\}$ as they have disjoint support. Thus, the collection $\{\boldsymbol{\phi}_{0,l}^1\}_{l=1:n_0^1} \cup \{\boldsymbol{\phi}_{1,l}^1\}_{l=1:n_1^1}$ forms an orthonormal basis for X_κ. From here, we apply this process recursively, generating an orthonormal basis of κ-Laplacian eigenvectors for each level in the given hierarchical bipartition tree.

We continue partitioning each κ-region until only singleton regions containing one κ-simplex remain; at this final level, the κ-HGLET vectors are simply the standard basis of $\mathbb{R}^n$. Each level of the dictionary contains an orthonormal basis (ONB) whose vectors have the support of generally half the size of the previous level, and the number of levels is generally logarithmic in n. There are roughly $(1.5)^n$ possible ONBs formed by selecting different covering sets of regions from the hierarchical bipartition tree; see, e.g., **[80, 93]** for more about the number of possible ONBs in such a hierarchical bipartition tree. Finally, we note that the computational cost of generating the entire dictionary is $\mathcal{O}(n^3)$ and that any valid hierarchical bipartition tree can be used to create a similar dictionary. Figure 5.1 shows the 2-HGLET constructed on the same 2-complex shown in Figure 4.7.

5.2. κ-Generalized Haar-Walsh Transform (κ-GHWT)

The second transform we present here is based on the *Generalized Haar-Walsh Transform* (GHWT) **[47]**, which can itself be viewed as a generalization of the Walsh-Hadamard transform. This basis dictionary is formed by first generating a hierarchical bipartition tree of C_κ. We then work in a bottom-up manner, beginning with the finest level in which each region only contains a

FIGURE 5.2. Coarse-to-Fine (C2F) 2-GHWT dictionary. The yellow, dark green, and violet regions in each vector indicate its positive, zero, and negative components, respectively.

single element. We call these functions scaling vectors and label them $\{\boldsymbol{\psi}_{k,0}^{j_{\max}}\}_{k=0:n\ \ 1}$. For the next level, we first assign a constant scaling vector for support on each region. Then, for each region that contains two children in the partition tree, we form a Haar-like basis element by subtracting the scaling function associated with the child element with a higher index from that child element with a lower index. This procedure will form an ONB $\{\boldsymbol{\psi}_{k,l}^{j_{\max}\ \ 1}\}_{k=0:k'\ \ 1,l=0:l(k)\ \ 1}$ where k' is the number of κ-subregions at level $j_{\max}\quad 1$ and $l(k) \in (1,2)$ depends on the partition k. The vectors of this basis have support of at most 2. For the next level, we begin by computing the scaling and Haar-like vectors as before. Next, for any region that contains three or more elements, we also compute Walsh-like vectors by adding and subtracting the Haar-like vectors in the children's regions. From here, we form the rest of the dictionary recursively. full description of this algorithm (for the $\kappa = 0$ case) is given in [**48**]. Figure 5.2 displays the 2-GHWT dictionary on the same 2-complex used in Figures 4.8 and 5.2. We make several observations about this dictionary. First, like the κ-HGLET, each level of the dictionary forms an ONB, and at each level, basis vectors have the support of roughly half the size of the previous level. These basis vectors also have the same support as the κ-HGLET basis vectors (that is, $\text{supp}(\boldsymbol{\phi}_{k,l}^{j}) = \text{supp}(\boldsymbol{\psi}_{k,l}^{j})$ for all j, k, l). However, the computational cost of computing the κ-GHWT is only $\mathcal{O}(n \log n)$ compared to the $\mathcal{O}(n^3)$ of the κ-HGLET.

Finally, we note that at the coarsest level ($j = 0$) the κ-GHWT dictionary contains globally-supported piecewise-constant basis vectors, which are ordered by increasing oscillation (or "sequency"). This forms an ONB analogous to the classical Walsh Basis, which we call the *κ-Walsh basis*. This allows us to define an associated Walsh transform and conduct Walsh analysis on signals defined on simplicial complexes.

5.3. Submatrix Partitioning

s mentioned in Section 2.3, we will use the submatrix method to hierarchically partition with whichever variety of κ-Laplacian Fiedler vector, so we will analyze how this scheme varies from rebuilding and partitioning with the restricted Laplacian on each subregion.

The analysis is most straightforward with the combinatorial L_κ. Let L_{full} denote the κ-Laplacian of the subregion C_k^j, and let L_{sub} refer to the submatrix of the usual κ-Laplacian of C_κ, induced by restricting the rows and columns to the simplex indices in C_k^j. Define S_{sub} analogously. For the lower Laplacian, notice that $L_{\text{full}}^\vee$ is exactly equal to the corresponding submatrix of $L_{\text{sub}}^\vee$, so no more work is necessary. Immediately, this means that if a κ-region contains no $(\kappa+1)$-hulls, then the hierarchical partitions by the two methods will be identical. For the upper Laplacian, if *any* of the $\kappa + 2$ faces of a given hull do not end up on the same side of a partition, then that hull is removed from both sides of the partition, changing both diagonal and non-diagonal entries of L_{full} relative to the corresponding submatrix of L_{sub}. If a subregion contains $0 < \ell < \kappa + 2$ of the faces of some hull in C_κ, then the submatrix partition will keep $\ell(\ell \quad 1)$ non-diagonal entries which would have been removed by the full partition, and the same corresponding contribution to the diagonal from the hull degrees. Explicitly, if $\sigma_1, \ldots, \sigma_\ell$ are these adjacent simplices in the same subregion, then for $i, j = 1, \ldots, \ell$,

$$[L_{\text{sub}}]_{\sigma_i\sigma_j} = \quad [S_{\text{sub}}]_{\sigma_i\sigma_j} + [L_{\text{full}}]_{\sigma_i\sigma_j} \quad \text{, and } \quad [L_{\text{sub}}]_{\sigma_i\sigma_i} = 1 + [L_{\text{full}}]_{\sigma_i\sigma_i} \ .$$

While this is no longer only a diagonal perturbation, it is explicitly a rank-one perturbation (per hull) which does not damage either the diagonal dominance of $L_{\text{sub}}^\wedge$, or its positive-semidefiniteness. Hence we expect the submatrix partitions to degrade in faithfulness to the full partitions at worst in proportion to the number of hulls lost to division by the partitions – gracefully at first, and

eventually completely. It is important to note that especially as κ increases, it becomes much more likely that any particular hull will be split across a partition. However, when the faces of that hull are split evenly across the partition, about half as many entries change for L_{full} as would have in an uneven split, and such even splits are expected; recall that S_κ is nonzero only for κ-adjacent simplices, not strongly adjacent ones.

The situation is more complicated for weighted κ-Laplacian variants, because with full partitioning, it may unexpectedly happen that because of the loss of a hull, then a simplex or boundary weight may become zero, and the entire basis computation may grind to a halt. t minimum, to prevent this situation, we must add weak adjacency weights, via the modified degree constraint (4.5). In lieu of an explicit perturbation analysis of the L_κ^{sym} case, we provide a suggestive experiment in Figure 5.3, depicting a series pairs of κ-GHWT basis vectors, where each pair contains a vector with the same (j, k, l) tags, but with one computed by full partitioning, and the other by submatrix partitioning. We use the naive Fiedler vector, instead of searching for a positive eigenvalue, because there is not a direct correspondence for this method in submatrix partitioning. This leads to lower quality partitions, but allows us to illustrate the properties of the submatrix partition. For small l-values, notice the distinct similarity between the partition regions. For the large l value, and higher level j, notice the graceful degradation to similarly sized and shaped, but different regions.

5.4. Basis Specification

For many downstream applications, it is important to organize the order of these bases. In general, the κ-HGLET dictionary is naturally ordered in a *Coarse-to-Fine* (C2F) fashion. In each region, the basis vectors are ordered by frequency (i.e., eigenvalue). Similarly, the GHWT dictionary is also naturally ordered in a C2F fashion, with increasing "sequency" within each subgraph. nother useful way to order the GHWT is in a *Fine-to-Coarse* (F2C) ordering, which approximates "sequency" domain partitioning. See, e.g., Figure 5.4, which shows the F2C 2-GHWT dictionary on the triangle graph. We also note that the F2C ordering is not possible for the κ-HGLET dictionary because some parent subspaces and the direct sum of their children subspaces are not equivalent; see, e.g., [**45**, Eq. (5.6)] for the details. Other relabeling schemes, such as those proposed in [**80**, **86**] may also be useful but are not explored here.

FIGURE 5.3. Comparison of submatrix partitioning with full partitioning, for a 2-region forming a closed triangular mesh. The dog toy mesh is from a Google Research dataset [**37**]. Each plot depicts a κ-GHWT basis vector, computed by either the full partitioning method (above) or the submatrix partitioning method (below). The (j, k, l) tags are indicated in the plot titles. In the colormap `viridis`, yellow indicates large positive values, purple large negative values, and green small-magnitude values.

Once we have established these arrangements of basis vectors, we can efficiently apply the best-basis algorithm [**20**] to select an ONB that is optimal for a task at hand for a given input signal or a class of input signals; see also related previous work of applying the best-basis algorithm for

FIGURE 5.4. Fine-to-Coarse (F2C) 2-GHWT dictionary. Note that this dictionary is not generated by simply reversing the row indices of the C2F dictionary, but instead by arranging each level (row) by "sequency".

hierarchical transforms in the graph setting $[\mathbf{17}, \mathbf{47}, \mathbf{48}, \mathbf{49}, \mathbf{51}, \mathbf{80}, \mathbf{86}]$. Given some cost function $\mathcal{F}$ and signal $\boldsymbol{x}$ (e.g., of coefficient values), we traverse the partition tree and select the basis that minimizes $\mathcal{F}$ restricted to each region. For the C2F dictionary, we initialize the best basis as the finest ($j = j_{\max}$) level of the GHWT dictionary. We then proceed upward one level at a time and compute the cost of each subspace at that level and compare it to the cost of the union of its children subspaces. If the latter cost is lower, the basis is updated; if not, the children subspaces (and their basis vectors) are propagated to the current level. This algorithm yields the C2F best basis. The F2C best basis is performed similarly, i.e., we begin with the globally-supported basis ($j = 0$) at the bottom of the rearranged tree and proceed in the same bottom-up direction. s for the HGLET dictionary, it has only a C2F basis as we discussed earlier. These properties pass identically to the κ-GHWT and κ-HGLET respectively.

In some contexts, it is not necessary to generate a complete ONB, but rather some sparse set of vectors in the dictionary (also known as atoms) that most accurately approximate a given signal or class of signals. In this case, we can directly apply the orthogonal matching pursuit of **[12]** to find the best m-dimensional orthogonal subframe ($m \leq n$) selected from the dictionary. dditionally, for some downstream tasks, such as sparse approximation or sparse feature selection, generating orthogonal sets of atoms is not critical. In these cases, we can employ a greedy algorithm to generate efficient approximation. This algorithm simply selects the atoms in the dictionary with the largest coefficient, removes it, then computes the transform of the residual and proceeds so forth.

CH PTER 6

pplications

We briefly explore the broad space of possible simplicial signal processing applications, with experiments using simulated and real-world data for approximation, compression, classification, and graph orientation. The graph orientation exposition and experiments are new, and the rest of the experiments either draw from or reproduce experiments from our preprint **[79]**. We will refer to functions on a κ-region as κ-signals.

6.1. pproximation and Signal Compression

We begin with an illustrative example on a synthetic dataset, by triangulating a digital image. We start with a 256×256 greyscale image of a woman from the USC-SIPI Image Database **[98]**, and map the pixel intensities of the image to a Cartesian grid on the unit square $[0, 1]^2$. We then sample 2000 points uniformly at random in the unit square, and construct a 2-complex from the Delaunay triangulation of these points. This simplicial complex contains 5955 edges and 3956 triangles. Next, we form κ-signals on this complex via interpolation. The 0-signal f at each vertex has the same value as the square of the Cartesian grid in which it is located. The 1-signal and 2-signal are each computed at a simplex σ as the average of the values $\{f(v) \mid v \in V(\sigma)\}$. typical result of this process is illustrated in Figure 6.1.

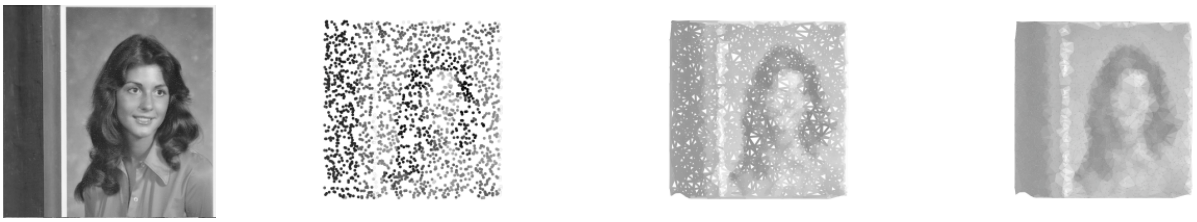


FIGURE 6.1. The construction of simplicial signals from an image. On the left is the greyscale image we start with. Next, the pixels are sampled, resulting in the second image. Then on the right are a 1-signal (on the edges) and 2-signal (on the triangles) on the simplicial complex formed by the Delaunay triangluation of these points.

Now, we construct sparse representations of these κ-signals using our multiscale bases. Figures 6.2 and 6.3 show the *nonlinear* approximation of the 1-signal and 2-signal, respectively, keeping those expansion coefficients which are largest in magnitude, via several bases: the pixel basis, the orthonormal κ-Walsh basis, the orthonormal κ-Haar basis, the κ-Fourier basis formed from the eigenvectors of L_κ, the κ-GHWT C2F best basis, and the κ-HGLET best basis. Figure 6.4 shows how the approximation error drops as the number of terms kept in each basis expansion grows.

number of observations are in order. First, the multiscale dictionary-based methods consistently outperformed the generic orthonormal bases. Overall the κ-GHWT-based method performed best, which match visual intuition that the coarse-scale features of this signal are the most important, and there are compression gains from simplifying the monotone background and simple geometric shapes. Similarly, the κ-Haar basis achieved much better results than the κ-Walsh basis, again emphasizing the significance of the coarser features. The κ-HGLET only found slightly better bases than the global Fourier basis in the $\kappa = 1$ case, suggesting that this signal was not generally smooth on subregions of the complex.

Next, we apply our approach to real-world data for higher κ-signals with $\kappa = 0, \ldots, 5$. The *coauthorship complex* **[30, 70]** is a simplicial complex derived from a citation dataset **[92]**, which models the interactions between multiple authors of scientific papers. Each paper has a number of citations, which are attributed to each of the paper's authors as follows. We first build a graph where each author is represented by a vertex, and an edge is present whenever two authors co-authored one or more papers together. Next, we form the clique complex of this graph, which is the domain on which we construct κ-signals. Then, for each κ, the citation κ-signal on a κ-simplex σ is the sum of the citation numbers for every paper such that each of the authors $v \in V(\sigma)$ are co-authors. See **[30]** for a more thorough description of the construction of this complex. Table 6.1 reports some basic information about the number of simplices of different degrees in this citation complex. Figure 6.5 shows the approximation of this signal (i.e., a vector of citation numbers) for $\kappa = 0, 1, \ldots, 5$ with the Delta, Fourier, κ-Haar, κ-HGLET, and κ-GHWT bases. Figure 6.6 shows the log error. The κ-HGLET and κ-GHWT bases were selected with the best-basis algorithm using the C2F ordering for the κ-GHWT dictionary.

FIGURE 6.2. Nonlinear approximation of an image-derived simplicial signal for $\kappa = 1$.

In these experiments, we observe that the best bases (κ-GHWT and κ-HGLET) outperformed the canonical bases, with the κ-GHWT being the most efficient basis for each κ. dditionally, for $\kappa > 0$, the orthonormal κ-Haar basis performed best in the semi-sparse regime (1 and 10% of terms retrained). This suggests that the signals on each degree of the citation complex are similar in that they are all close to being piecewise constant. However, when more terms are considered, the κ-HGLET best basis achieved a lower approximation error than the orthonormal Haar basis achieved.

FIGURE 6.3. Nonlinear approximation of an image-derived simplicial signal for $\kappa = 2$.

κ	0	1	2	3	4	5
# of elements	1126	5059	11840	18822	21472	17896

TABLE 6.1. The number of element in the κ-simplices in the coauthorship complex for $\kappa = 0, 1, \ldots, 5$

6.2. Signal Clustering and Classification

Since the basis (and dictionary) vectors we present are both multiscale and built from κ-Laplacians that are aware of both topological and geometric properties of the domain [**14**], they can function as very powerful feature extractors for general data science applications. In this section, we present two basic classification experiments in the $\kappa = 1$ setting, in which we modify well-known

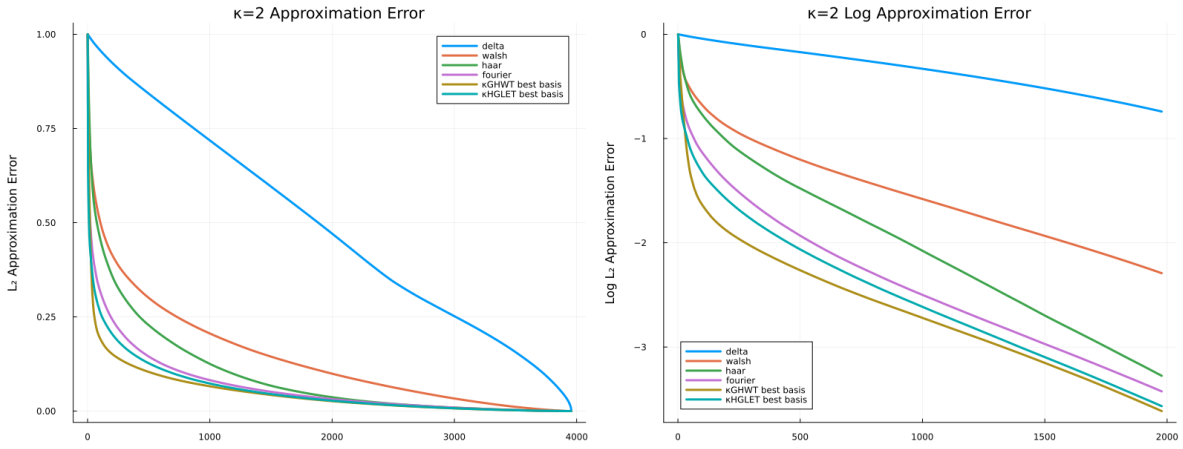


FIGURE 6.4. Nonlinear approximation errors for the image-derived simplicial signal, with L_2 error on the left, and $\log(L_2)$ error for up to half of terms retained on the right. The top diagrams show $\kappa = 1$, and the bottom $\kappa = 2$.

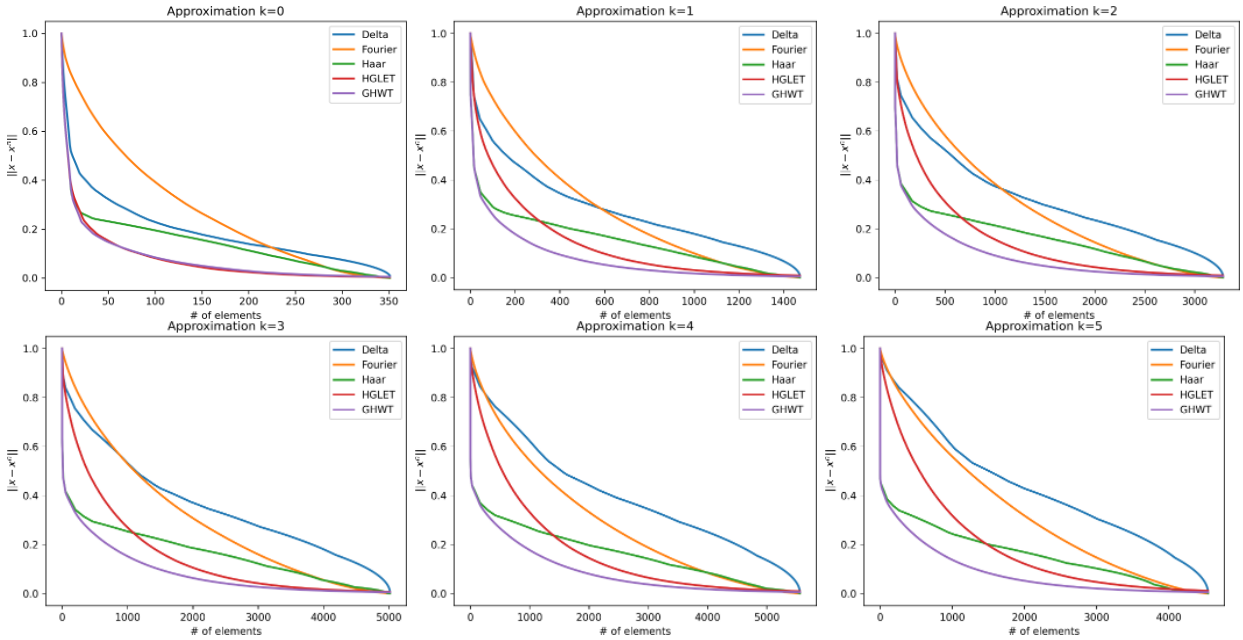


FIGURE 6.5. pproximation of the Citation Complex for $\kappa = 0, \ldots, 5$.

image datasets to create 1-simplicial signals, in order to demonstrate the effectiveness of our basis dictionaries for representing these signals. s a baseline and sanity check, we compare our proposed dictionaries with the Fourier basis $\{\boldsymbol{\phi}^l_{0,0}\}_l$ composed of global L_κ-eigenvectors, and the Delta or "pixel" basis $\{\boldsymbol{\psi}^0_{j_{\max},k}\}_k$. Further, we compare our results with those achieved by the Joint and

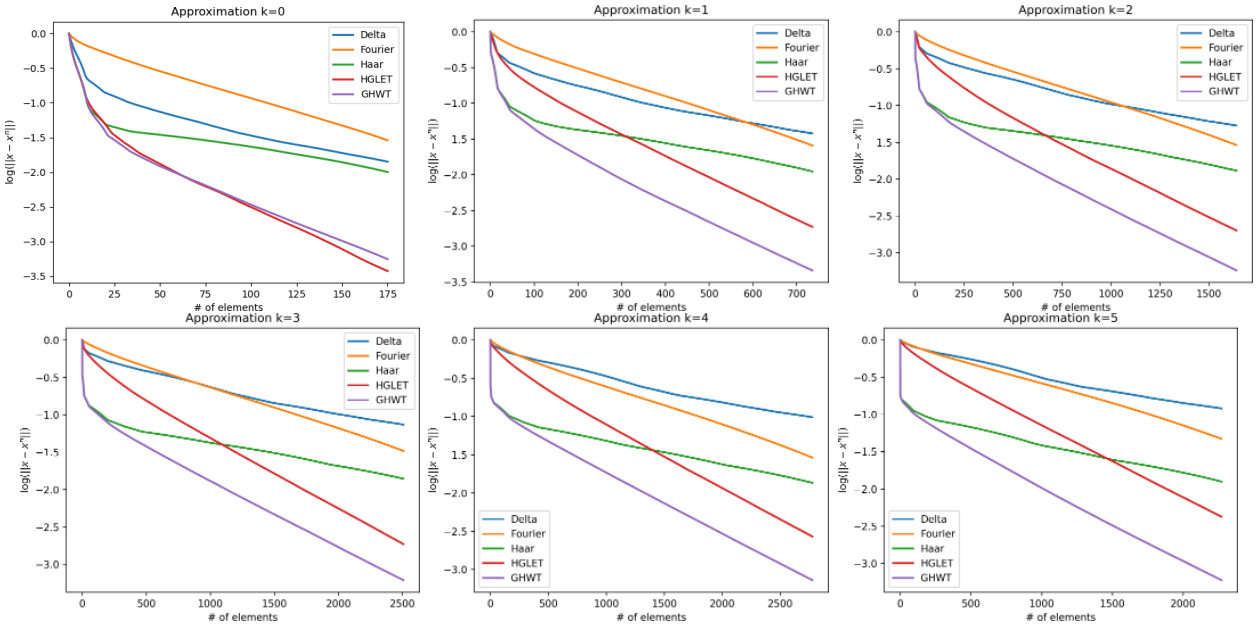


FIGURE 6.6. Top: pproximation of the Citation Complex for $\kappa = 0, \ldots, 5$. Bottom: Log of the error for up to 50% of the terms retained.

Separate *Hodgelets* proposed in **[72]**, which are wavelet transforms on **1**-simplices constructed from the eigenvectors of L , and separately from $L_1^{\wedge}, L_1^{\vee}$ respectively.

For our first experiment, we construct a dataset of 1-signals using 1000 handwritten digits from the MNIST dataset **[63]** by sampling 500 points in the unit square and following the image interpolation method presented in Section 6.1. We then compute the features of these images using the proposed orthogonal transforms and best bases from the overcomplete dictionaries. Next, we train a support vector machine (SVM) **[39]** to classify the digits for each of the transformed representations using the 1000 training examples. Finally, we test these SVMs on the rest of the entire MNIST dataset.

We repeat this experiment for the FMNIST dataset **[100]**, again using 1000 examples for training data. Results are presented in Table 6.2. We remark that these tests are not meant to achieve state-of-the-art results for image classification but rather to showcase the effectiveness of these representations for downstream tasks. Unsurprisingly, the dictionary methods outperformed the basis methods. gain, the piecewise constant methods (κ-GHWT, κ-Haar) achieved better approximations than the smoother methods (Fourier, κ-HGLET, Joint, and Separate Hodgelets). This is likely due to the near-binary nature of images in both datasets.

	Basis Methods							Dictionary Methods			
	Delta	Fourier	κ-Haar	κ-Walsh	κ-HGLET (BB)	κ-GHWT (BB C2F)	κ-GHWT (BB F2C)	Joint	Separate	κ-HGLET	κ-GHWT
# of terms	661	661	661	661	661	661	661	5288	5288	9254	9254
MNIST	68.675	77.053	75.388	77.011	77.991	**78.779**	77.156	79.202	80.038	80.001	**81.089**
FMNIST	64.370	76.753	**7**6.779	75.230	76.117	**76.991**	76.121	78.761	78.738	79.739	**80.789**

TABLE 6.2. Test ccuracy for SVMs trained on transforms of MNIST signals interpolated to a random triangulation.

6.3. Graph Orientation

Considering our interpretation in Section 2.4 of oriented 1-simplices as directed edges, and the invariance of the Fiedler vector for κ-regions to simplex orientation, it is clear that there is a close relationship between the 1-Laplacian and directed structures on graphs. For example, in problems of *graph orientation*, one wishes to convert a (possibly weighted) graph G into a directed graph , by adding a direction to each edge, in order to fulfill some structural objective, like minimizing the total weighted in-degree of resulting sinks [**4**], or minimizing the maximum weighted out-degree among all vertices [**3**]. Equivalently, we may ask, for each assignment of natural or reverse orientation to the 1-simplices of $K_1(G)$, which one best fulfills the objective? Recall that the eigenvectors of L_1 are affected in a simple way by changes in orientation to the underlying simplices: a flip in orientation to some simplex yields a sign change of the corresponding eigenvector coordinate. This property makes the eigenvectors of L_1 an attractive target for solving relaxed versions of combinatorial optimization problems over assignments of directions to the edges of G. We will not prove a theorem, but illustrate why $\boldsymbol{\phi}_{\mathfrak{b}_\kappa+1}(L_1)$ appears to find good solutions to some graph optimization problems.

Recall the Hodge decomposition Eq. (4.2); utilizing the language of [**65**], for $\kappa = 1$ we can think of B_0 as the discrete *divergence* operator, and B_1 as the discrete *curl* operator, so that $\ker(L_1^\vee)$ describes divergence-free 1-signals, and $\ker(L_1^\wedge)$ describes curl-free 1-signals. For an eigenpair $(\lambda, \boldsymbol{\phi})$ of L_1 where $\lambda > 0$, exactly one of these will be the the case, so that the signal $\boldsymbol{\phi}$ is either a pure gradient, or pure co-curl. Figure 6.7 demonstrates the former case, as it depicts a graph G, in particular a tree, with $L_1^\wedge =$, and in the middle plot the bottom eigenvector $\boldsymbol{\phi}_0$ is depicted, with $\lambda_0 > 0$. This graph's structure depicts the dendrites of a neuron. The left-hand plot is a visualization of the vertex indexing used in constructing this graph, which demonstrates the key

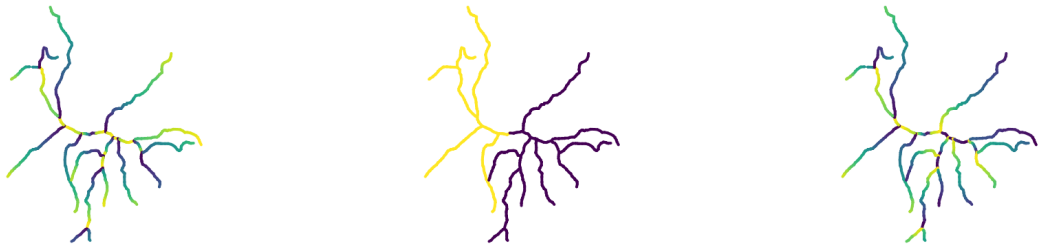


FIGURE 6.7. n example of graph orientation, performed on a dendritic tree. On the left, we show the natural orientation of the edges in the graph. We plot a smooth gradient on each segment of the tree, such that increasing vertex index is mapped to the change from purple to yellow in the usual `viridis` colormap. In the middle is the sign of $\boldsymbol{\phi}_1(L_1)$, plotted on the edges. On the right are the orientations given by flipping edges where $\boldsymbol{\phi}_1(L_1)$ is negative (so, purple in the middle plot).

fact that along each segment of the tree, the vertices are sequentially indexed, with the sequence increasing *away* from a central vertex, so that the natural orientation of each edge is thus also away from the central vertex. The middle plot then can be interpreted as an assignment of direction to each edge of G, such that edges to the left of the central vertex remain in natural orientation while the rest are flipped, or vice versa. This re-orientation, depicted in the plot on the right, is then certainly an improved gradient flow, in the sense that a directed path can be followed from any vertex to the right of the central vertex, to any vertex to the left of it, whereas in the natural orientation, directed paths can only be followed away from the central vertex. Eigenvectors $\boldsymbol{\phi}_i$ for $i > 0$ with larger eigenvalue reproduce this effect, only less effectively, with fewer possible directed paths.

Now, recall from Section 4.2 that the bottom non-harmonic eigenvector $\boldsymbol{\phi}_{\mathfrak{b}_\kappa+1}$ of L_1 acts as a relaxed minimizer of κCut, one characterization of which is maximizing the number of pairs of consistent adjacent simplices. Consider many edges meeting at a vertex. Under what kinds of orientations are the consistency of these pairs of edges maximized? We must orient equally, or as close to equally as possible, many edges adjacent to this vertex to treat it as a head, and as a tail; in other words, to balance the vertex's in-degree and out-degree. Finding an orientation for a graph which minimizes the sum of differences between in-degree and out-degree across all vertices is known to be NP-hard in general [**57**], and this eigenvector appears to provide a relaxed solution.

CH PTER 7

Conclusion

In this dissertation, we have presented two examples of multiscale transforms for analyzing signals on κ-regions of simplicial complexes, that generalize multiscale transforms previously defined only on graphs: the Hierarchical κ-Laplacian Eigen Transform (κ-HGLET) and κ-Generalized Haar-Walsh Transform. These generalizations both provide on the one hand higher-order analysis for signals, and of higher-order relationship, and on the other, analysis for signals with underlying oriented structure, potentially providing new perspective and avenues for related areas, like directed graphs, or non-alternating signals on oriented simplicial complexes. We have illustrated the parallels between commuting discrete integral and differential operators, and their continuous analogs, opening new directions into spectral geometry and clustering, and additional tools for partitioning of graphs, in service of the multiscale transforms described. Working with data on simplicial complexes induces combinatorial complexity, and requires a computationally efficient approach at scale, so along with our numerical experiments, we carefully developed code backed by a novel interpretation of the κ-Laplacian to construct efficient data structures for computing and storing adjacency and consistency information for κ-regions of simplicial complexes. The tree data structures involved are analogous to those of the Multiscale Transforms for Signals on Graphs (MTSG) toolbox, introduced in [**44**] and expanded in Julia in [**46**], and we use these Multiscale Transforms for Signals on Simplices (MTSS) for efficient analysis and synthesis via the κ-signal transforms introduced.

Many further topics remain to explore. critical but yet elusive topic is the unification of the discrete integral operator perspective with the κ-simplex perspective: understanding the appropriate notion of distance over κ-regions which acts to generalize the harmonic kernel, and from which spectral analysis of simplicial complexes can yield new geometric and/or topological information. We have begun developing an analogous theory of Green's functions for the vector Poisson equation, and the discretization of integral operators involving differential forms and their reduction to linear

algebra over the k-simplices of an appropriate complex, which we hope to successfully relate to the spectral theory of the Hodge Laplacian.

Then, there are more ingredients to add to the κ-simplex extensions of these multiscale transforms, that have been well-developed for the graph setting. Entire families of such multiscale transforms include the *extended Generalized Haar-Walsh Transform*(eGHWT) [**86**] and *Natural Graph Wavelet Packets* (NGWPs) [**17**]. Further, we may explore different best-basis selection criteria tailored for classification and regression problems such as the *Local Discriminant asis* [**76**, **78**] and the *Local Regression asis* [**77**] on simplicial complexes. Finally, it appears fruitful to investigate nonlinear feature extraction techniques such as the *Geometric Scattering Transform* [**34**] in the simplicial setting.

There is an abundance of exciting applications that open up in the simplicial setting, and especially for analysis of trajectories and directed graphs. These include data science problems in computational chemistry, weather forecasting, genetic analysis, social network analysis, and financial modeling, all of which have elements that are naturally modeled with simplicial complexes.

Bibliography

[1] R. . dams and J. J. Fournier, *Sobolev Spaces*, Elsevier, 2nd ed., 2003.

[2] S. S. gaian, H. Sarukhanyan, K. Egiazarian, and J. stola, *Hadamard Transforms*, SPIE, 2011.

[3] Y. sahiro, E. Miyano, H. Ono, and K. Zenmyo, *Graph orientation algorithms to minimize the maximum outdegree*, International Journal of Foundations of Computer Science, 18 (2007), pp. 197–215.

[4] M. J. tallah, *graph orientation problem*, Department of Computer Science Technical Reports, (1983).

[5] R. Bapat, S. J. Kirkland, and M. Neumann, *On distance matrices and Laplacians*, Linear lgebra and Its pplications, 401 (2005), pp. 193–209.

[6] R. B. Bapat and T. Raghavan, *Nonnegative Matrices and pplications*, Cambridge University Press, 1997.

[7] S. Barbarossa and S. Sardellitti, *Topological signal processing over simplicial complexes*, IEEE Trans. Signal Process., 68 (2020), pp. 2992–3007.

[8] M. Belkin and P. Niyogi, *Laplacian eigenmaps for dimensionality reduction and data representation*, Neural Computation, 15 (2003), pp. 1373–1396.

[9] C. Berge, *Hypergraphs: Combinatorics of Finite Sets*, vol. 45, Elsevier, 1984.

[10] J.-D. Boissonnat and C. Maria, *The simplex tree: n efficient data structure for general simplicial complexes*, lgorithmica, 70 (2014), pp. 406–427.

[11] J. Bruna, W. Zaremba, . Szlam, and Y. LeCun, *Spectral networks and locally connected networks on graphs*, in International Conference on Learning Representations (ICLR2014), CBLS, pril 2014, 2014.

[12] T. T. Cai and L. Wang, *Orthogonal matching pursuit for sparse signal recovery with noise*, IEEE Trans. Inform. Theory, 57 (2011), pp. 4680–4688.

[13] G. Carlsson, *Topology and data*, Bull. mer. Math. Soc., 46 (2009), pp. 255–308.

[14] Y.-C. Chen, M. Meilă, and I. G. Kevrekidis, *Helmholtzian eigenmap: Topological feature discovery & edge flow learning from point cloud data*, arXiv preprint arXiv:2103.07626, (2021).

[15] F. Chung and L. Lu, *Complex Graphs and Networks*, no. 107 in CBMS Regional Conference Series in Mathematics, mer. Math. Soc., Providence, RI, 2006.

[16] F. R. Chung, *Spectral Graph Theory*, vol. 92, merican Mathematical Soc., 1997.

[17] . Cloninger, H. Li, and N. Saito, *Natural graph wavelet packet dictionaries*, J. Fourier nal. ppl., 27 (2021), p. 41. a part of "Topical Collection: Harmonic nalysis on Combinatorial Graphs".

[18] R. R. Coifman and Y. Meyer, *Nouvelles bases orthonormées de L^2 $\mathbb{R}$) ayant la structure du systèm de Walsh*, preprint, Dept. of Mathematics, Yale University, New Haven, CT, ug. 1989.

[19] R. R. Coifman, Y. Meyer, and V. Wickerhauser, *Wavelet analysis and signal processing*, in Wavelets and Their pplications, M. Ruskai et al., eds., Jones and Bartlett, Boston, 1992, pp. 153–178.

[20] R. R. Coifman and M. V. Wickerhauser, *Entropy-based algorithms for best basis selection*, IEEE Trans. Inform. Theory, 38 (1992), pp. 713–718.

[21] J. W. Cooley and J. W. Tukey, *n algorithm for the machine calculation of complex Fourier series*, Mathematics of computation, 19 (1965), pp. 297–301.

[22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to lgorithms*, MIT Press, 4th ed., 2022.

[23] K. Crane, *Discrete differential geometry: n applied introduction*, Notices of the MS, Communication, (2018), pp. 1153–1159.

[24] M. Cucuringu, P. Davies, . Glielmo, and H. Tyagi, *SPONGE: generalized eigenproblem for clustering signed networks*, in The 22nd International Conference on rtificial Intelligence and Statistics, PMLR, 2019, pp. 1088–1098.

[25] I. Daubechies, *Ten Lectures on Wavelets*, SI M, 1992.

[26] E. B. Davies, G. M. L. Gladwell, J. Leydold, and P. F. Stadler, *Discrete nodal domain theorems*, Linear lgebra ppl., 336 (2001), pp. 51–60.

[27] C. Ding, X. He, and H. D. Simon, *On the equivalence of nonnegative matrix factorization and spectral clustering*, in Proceedings of the 2005 SI M International Conference on Data Mining, SI M, 2005, pp. 606–610.

[28] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, *Graph signal processing for machine learning: review and new perspectives*, IEEE Signal Processing Magazine, 37 (2020), pp. 117–127.

[29] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning and a Highly Connected World*, Cambridge Univ. Press, New York, 2010.

[30] S. Ebli, M. Defferrard, and G. Spreemann, *Simplicial neural networks*, arXiv preprint arXiv:2010.03633, (2020).

[31] M. Fiedler, *property of eigenvectors of nonnegative symmetric matrices and its application to graph theory*, Czechoslovak Mathematical Journal, 25 (1975), pp. 619–633.

[32] G. B. Folland, *Introduction to Partial Differential Equations*, Princeton University Press, 2nd ed., 1995.

[33] J. Friedman, *Computing Betti numbers via combinatorial Laplacians*, in Proceedings of the 28th nnual CM Symposium on Theory of Computing, 1996, pp. 386–391.

[34] F. Gao, G. Wolf, and M. Hirn, *Geometric scattering for graph data analysis*, in International Conference on Machine Learning, PMLR, 2019, pp. 2122–2131.

[35] C. Giusti, R. Ghrist, and D. S. Bassett, *Two's company, three (or more) is a simplex*, J. Comput. Neurosci., 41 (2016), pp. 1–14.

[36] D. Gleich, *MatlabBGL – a Matlab Graph Library*. https://www.cs.purdue.edu/homes/dgleich/packages/matlab_bgl/index.html, 2006.

[37] Google Research, *Toysmith Windem Up Flippin nimals Dog*. https://fuel.gazebosim.org/1.0/GoogleResearch/models/Toysmith_Windem_Up_Flippin_ nimals_Dog, September 2020.

[38] L. Hagen and . B. Kahng, *New spectral methods for ratio cut partitioning and clustering*, IEEE Trans. Comput.- ided Des., 11 (1992), pp. 1074–1085.

[39] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2, Springer, 2009.

[40] L. Hermi and N. Saito, *On Rayleigh-type formulas for a non-local boundary value problem associated with an integral operator commuting with the Laplacian*, pplied and Computational Harmonic nalysis, 45 (2018), pp. 59–83.

[41] M. Holzrichter and S. Oliveira, *graph based method for generating the Fiedler vector of irregular problems*, in International Parallel Processing Symposium, Springer, 1999, pp. 978–985.

[42] R. . Horn and C. R. Johnson, *Topics in Matrix nalysis*, Cambridge University Press, 1994.

[43] ———, *Matrix nalysis*, Cambridge University Press, 2nd ed., 2012.

[44] J. Irion, *Github - JeffLIrion/MTSG_ Toolbox: The Multiscale Transforms for Signals on Graphs (MTSG) Toolbox*. https://github.com/JeffLIrion/MTSG_Toolbox. [ccessed 11-Jan-2022].

[45] ———, *Multiscale Transforms for Signals on Graphs: Methods and pplications*, PhD thesis, ppl. Math., Univ. California, Davis, Dec. 2015.

[46] J. Irion, H. Li, N. Saito, and Y. Shao, *Multiscale graph signal transforms*. https://github.com/UCD4IDS/MultiscaleGraphSignalTransforms.jl, 2021.

[47] J. Irion and N. Saito, *The generalized Haar-Walsh transform*, in 2014 IEEE Workshop on Statistical Signal Processing (SSP), IEEE, 2014, pp. 472–475.

[48] ———, *Hierarchical graph Laplacian eigen transforms*, JSI M Letters, 6 (2014), pp. 21–24.

[49] ———, *pplied and computational harmonic analysis on graphs and networks*, in Wavelets and Sparsity XVI, Proc. SPIE 9597, M. Papadakis, V. K. Goyal, and D. Van De Ville, eds., 2015. Paper # 95971F.

[50] ———, *Learning sparsity and structure of matrices with multiscale graph basis dictionaries*, in Proc. 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), . Uncini, K. Diamantaras, F. . N. Palmieri, and J. Larsen, eds., 2016.

[51] ———, *Efficient approximation and denoising of graph signals using the multiscale basis dictionaries*, IEEE Trans. Signal Inform. Process. Netw., 3 (2017), pp. 607–616.

[52] S. Jaffard, Y. Meyer, and R. D. Ryan, *Wavelets: Tools for Science & Technology*, SI M, Philadelphia, P , 2001.

[53] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye, *Statistical ranking and combinatorial Hodge theory*, Math. Program., 127 (2011), pp. 203–244.

[54] F. John, *Partial Differential Equations*, vol. 1 of pplied Mathematical Sciences, Springer-Verlag, New York, 4th ed., 1982.

[55] D. Joncas, M. Meila, and J. McQueen, *Improved graph Laplacian via geometric self-consistency*, in dvances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., vol. 30, Curran ssociates, Inc., 2017.

[56] G. Kaiser and L. H. Hudgins, *Friendly Guide to Wavelets*, vol. 300, Springer, 1994.

[57] J. Kára, J. Kratochvil, and D. R. Wood, *On the complexity of the balanced vertex ordering problem*, Discrete Mathematics & Theoretical Computer Science, Vol. 9 no. 1 (2007).

[58] . V. Knyazev, *Signed Laplacian for spectral clustering revisited*. `https://www.merl.com/publications/docs/TR2017-001.pdf`, 2017. Mitsubishi Electric Research Laboratories.

[59] S. J. Koelle, H. Zhang, M. Meila, and Y.-C. Chen, *Manifold coordinates with physical meaning*, Journal of Machine Learning Research, 23 (2022), pp. 1–57.

[60] R. Kondor and S. Trivedi, *On the generalization of equivariance and convolution in neural networks to the action of compact groups*, in International Conference on Machine Learning, PMLR, 2018, pp. 2747–2755.

[61] R. Kress, *Linear Integral Equations*, vol. 82 of pplied Mathematical Sciences, Springer-Verlag, New York, 3rd ed., 2014.

[62] J. Kunegis, S. Schmidt, . Lommatzsch, J. Lerner, E. W. D. Luca, and S. lbayrak, *Spectral analysis of signed graphs for clustering, prediction and visualization*, in Proceedings of the 2010 SI M International Conference on Data Mining (SDM), SI M, 2012, pp. 559–570.

[63] Y. LeCun and C. Cortes, *MNIST handwritten digit database*. `http://yann.lecun.com/exdb/mnist/`, 2010.

[64] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *RP CK Users' Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted rnoldi Methods*, SI M, 1998.

[65] L.-H. Lim, *Hodge Laplacians on graphs*, SI M Review, 62 (2020), pp. 685–715.

[66] L. Lovász, *Large Networks and Graph Limits*, vol. 60 of Colloquium Publications, mer. Math. Soc., Providence, RI, 2012.

[67] S. Mallat, *Wavelet Tour of Signal Processing*, cademic Press, Burlington, M , 3rd ed., 2009.

[68] M. Newman, *Networks*, Oxford Univ. Press, Oxford, UK, 2nd ed., 2018.

[69] . Ortega, P. Frossard, J. Kovacević, J. M. Moura, and P. Vandergheynst, *Graph signal processing: Overview, challenges, and applications*, Proc. IEEE, 106 (2018), pp. 808–828.

[70] . Patania, G. Petri, and F. Vaccarino, *The shape of collaborations*, EPJ Data Science, 6 (2017), pp. 1–16.

[71] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.

[72] T. M. Roddenberry, F. Frantzen, M. T. Schaub, and S. Segarra, *Hodgelets: Localized spectral representations of flows on simplicial complexes*, in IC SSP 2022-2022 IEEE International Conference on coustics, Speech and Signal Processing (IC SSP), IEEE, 2022, pp. 5922–5926.

[73] T. M. Roddenberry, M. T. Schaub, and M. Hajij, *Signal processing on cell complexes*, in IC SSP 2022-2022 IEEE International Conference on coustics, Speech and Signal Processing (IC SSP), IEEE, 2022, pp. 8852–8856.

[74] N. Saito, *Data analysis and representation on a general domain using eigenfunctions of Laplacian*, pplied and Computational Harmonic nalysis, 25 (2008), pp. 68–97.

[75] ———, *How can we naturally order and organize graph Laplacian eigenvectors?*, in Proc. 2018 IEEE Workshop on Statistical Signal Processing, 2018, pp. 483–487.

[76] N. Saito and R. R. Coifman, *Local discriminant bases and their applications*, J. Math. Imaging Vis., 5 (1995), pp. 337–358. Invited paper.

[77] ———, *Extraction of geological information from acoustic well-logging waveforms using time-frequency wavelets*, Geophysics, 62 (1997), pp. 1921–1930.

[78] N. Saito, R. R. Coifman, F. B. Geshwind, and F. Warner, *Discriminant feature extraction using empirical probability density estimation and a local basis library*, Pattern Recognition, 35 (2002), pp. 2841–2852.

[79] N. Saito, S. C. Schonsheck, and E. Shvarts, *Multiscale transforms for signals on simplicial complexes*. `https://arxiv.org/abs/2301.02136`, 2023. arXiv.

[80] N. Saito and Y. Shao, *eGHWT: The Extended Generalized Haar–Walsh Transform*, J. Math. Imaging Vis., 64 (2022), pp. 261–283.

[81] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, Inc., San Francisco, C , 3rd ed., 2006.

[82] M. T. Schaub, . R. Benson, P. Horn, G. Lippner, and . Jadbabaie, *Random walks on simplicial complexes and the normalized Hodge 1-Laplacian*, SI M Review, 62 (2020), pp. 353–391.

[83] M. T. Schaub, Y. Zhu, J.-P. Seby, T. M. Roddenberry, and S. Segarra, *Signal processing on higher-order networks: Livin' on the edge ... and beyond*, Signal Processing, 187 (2021), p. 108149.

[84] N. C. Schonsheck and S. C. Schonsheck, *Spherical coordinates from persistent cohomology*, arXiv preprint arXiv:2209.02791, (2022).

[85] S. C. Schonsheck, B. Dong, and R. Lai, *Parallel transport convolution: Deformable convolutional networks on manifold-structured data*, SI M Journal on Imaging Sciences, 15 (2022), pp. 367–386.

[86] Y. Shao and N. Saito, *The extended generalized Haar-Walsh transform and applications*, in Wavelets and Sparsity XVIII, Proc. SPIE 11138, Paper #111380C, 2019.

[87] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Trans. Pattern nal. Machine Intell., 22 (2000), pp. 888–905.

[88] D. I. Shuman, S. K. Narang, P. Frossard, . Ortega, and P. Vandergheynst, *The emerging field of signal processing on graphs*, IEEE Signal Processing Magazine, 30 (2013), pp. 83–98.

[89] E. M. Stein and R. Shakarchi, *Fourier nalysis: n Introduction*, Princeton University Press, 2011.

[90] S. Steinerberger, *The first eigenvector of a distance matrix is nearly constant*, Discrete Mathematics, 346 (2023), p. 113291.

[91] G. Strang, *The discrete cosine transform*, SI M Review, 41 (1999), pp. 135–147.

[92] L. Subelj and M. Bajec, *Model of complex networks based on citation dynamics*, in Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 527–530.

[93] C. M. Thiele and L. F. Villemoes, *fast algorithm for adapted time-frequency tilings*, ppl. Comput. Harmon. nal., 3 (1996), pp. 91–99.

[94] J. L. Troutman, *The logarithmic potential operator*, Illinois Journal of Mathematics, 11 (1967), pp. 365–374.

[95] P. J. Van Fleet and D. K. Ruch, *Wavelet Theory: n Elementary pproach with pplications*, John Wiley & Sons, 2011.

[96] U. von Luxburg, *tutorial on spectral clustering*, Stat. Comput., 17 (2007), pp. 395–416.

[97] U. Von Luxburg, . Radl, and M. Hein, *Hitting and commute times in large random neighborhood graphs*, The Journal of Machine Learning Research, 15 (2014), pp. 1751–1798.

[98] . Weber, *USC-SIPI Image Database*. `https://sipi.usc.edu/database/`.

[99] M. V. Wickerhauser, *dapted Wavelet nalysis from Theory to Software*, K Peters, Ltd., Wellesley, M , 1994.

[100] H. Xiao, K. Rasul, and R. Vollgraf, *Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms*, arXiv preprint arXiv:/1708.07747, (2017).

[101] X. Xue, *On a Fast lgorithm for Computing the Laplacian Eigenpairs via Commuting Integral Operators*, PhD thesis, Dept. Math., Univ. California, Davis, 2007.

[102] . Zomorodian, *Fast construction of the Vietoris-Rips complex*, Computers & Graphics, 34 (2010), pp. 263–271.