Tutorial
Harmonic Analysis *on* and *of* Irregular Domains, Graphs, and Networks

Naoki Saito

Department of Mathematics
University of California, Davis

2014 IEEE Workshop on Statistical Signal Processing
Gold Coast, AUSTRALIA
June 29, 2014

Outline

Acknowledgment

- Mark Ashbaugh (Univ. Missouri)
- Pedro Antunes (Univ. Lisbon, Portgal)
- Alex Barnett (Dartmouth College)
- Faisal Beg & Pradeep Kumar (Simon Fraser Univ., Canada)
- Raphy Coifman & Peter Jones (Yale Univ.)
- David Donoho (Stanford Univ.)
- Lotfi Hermi (Univ. Arizona)
- John Hunter (UC Davis)
- Hrushikesh Mhaskar (Claremont Grad. Univ.)
- Braxton Osting (UCLA $\longrightarrow$ Univ. Utah)
- Martin Reuter (Harvard Medical School)
- My current & former students at UC Davis
- Support from NSF & ONR
- The MacTutor History of Mathematics Archive, Wikipedia, ...

General Basic References

- For irregular domains:
  - W. A. Strauss: *Partial Differential Equations: An Introduction*, 2nd Ed., Chap. 10 & 11, John Wiley & Sons, 2009.
  - R. Courant & D. Hilbert: *Methods of Mathematical Physics*, Vol. I, Chap. V, VI, & VII, Wiley-Interscience, 1953.
  - D. S. Grebenkov & B.-T. Nguyen: "Geometrical structure of Laplacian eigenfunctions," *SIAM Review*, vol. 55, no. 4, pp.601–667, 2013
  - `http://www.math.ucdavis.edu/~saito/courses/LapEig/refs.html`
- For graphs and networks:
  - R. B. Bapat: *Graphs and Matrices*, Universitext, Springer, 2010.
  - F. R. K. Chung: *Spectral Graph Theory*, Amer. Math. Soc., 1997.
  - D. Cvetković, P. Rowlinson, & S. Simić: *An Introduction to the Theory of Graph Spectra*, Vol. 75, London Mathematical Society Student Texts, Cambridge Univ. Press, 2010.
  - D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, & P. Vandergheynst: "The emerging field of signal processing on graphs," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
  - `http://www.math.ucdavis.edu/~saito/courses/HarmGraph/refs.html`
- Specific references are given throughout the lectures.

Outline

Outline

1. Motivations
   - Motivations: Why Irregular Domains?
   - Motivations: Why Graphs?
2. History of Laplacian Eigenvalue Problems – Spectral Geometry
3. Harmonic Analysis of/on Irregular Domains via Eigenfunctions of Integral Operators Commuting with Laplacians
4. Harmonic Analysis of/on Graphs & Networks via Graph Laplacians
5. Summary & References

Motivations: Why Irregular Domains?

- Consider a bounded domain of general (may be quite complicated) shape $\Omega \subset \mathbb{R}^d$.
- Want to analyze the spatial frequency information inside of the object defined in $\Omega \implies$ need to avoid the Gibbs phenomenon due to $\partial\Omega$.
- Want to represent the object information efficiently for analysis, interpretation, discrimination, etc. $\implies$ need fast decaying expansion coefficients relative to a meaningful basis.
- Want to extract geometric information about the domain $\Omega \implies$ shape clustering/classification.

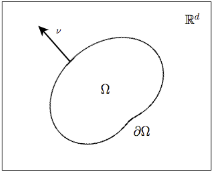


Figure: $\Omega \subset \mathbb{R}^d$ with ν being a normal vector on $\partial\Omega$.

Motivations: Why Irregular Domains?

- Consider a bounded domain of general (may be quite complicated) shape $\Omega \subset \mathbb{R}^d$.
- Want to analyze the spatial frequency information inside of the object defined in $\Omega \implies$ need to avoid the Gibbs phenomenon due to $\partial\Omega$.
- Want to represent the object information efficiently for analysis, interpretation, discrimination, etc. $\implies$ need fast decaying expansion coefficients relative to a meaningful basis.
- Want to extract geometric information about the domain $\Omega \implies$ shape clustering/classification.

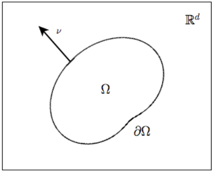


Figure: $\Omega \subset \mathbb{R}^d$ with ν being a normal vector on $\partial\Omega$.

Motivations: Why Irregular Domains?

- Consider a bounded domain of general (may be quite complicated) shape $\Omega \subset \mathbb{R}^d$.
- Want to analyze the spatial frequency information inside of the object defined in $\Omega \Longrightarrow$ need to avoid the Gibbs phenomenon due to $\partial\Omega$.
- Want to represent the object information efficiently for analysis, interpretation, discrimination, etc. $\Longrightarrow$ need fast decaying expansion coefficients relative to a meaningful basis.
- Want to extract geometric information about the domain $\Omega \Longrightarrow$ shape clustering/classification.

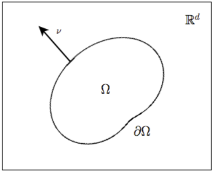


Figure: $\Omega \subset \mathbb{R}^d$ with ν being a normal vector on $\partial\Omega$.

Motivations: Why Irregular Domains?

- Consider a bounded domain of general (may be quite complicated) shape $\Omega \subset \mathbb{R}^d$.
- Want to analyze the spatial frequency information inside of the object defined in $\Omega \Longrightarrow$ need to avoid the Gibbs phenomenon due to $\partial\Omega$.
- Want to represent the object information efficiently for analysis, interpretation, discrimination, etc. $\Longrightarrow$ need fast decaying expansion coefficients relative to a meaningful basis.
- Want to extract geometric information about the domain $\Omega \Longrightarrow$ shape clustering/classification.

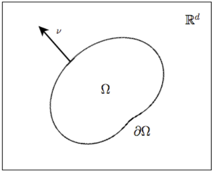
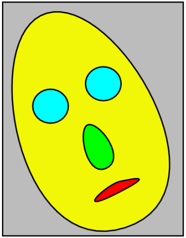
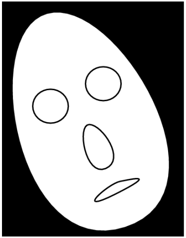


Figure: $\Omega \subset \mathbb{R}^d$ with ν being a normal vector on $\partial\Omega$.

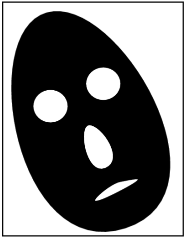
Object-Oriented Image Analysis



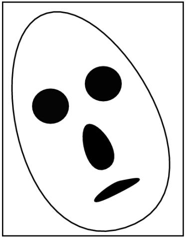
(a) Original



(b) Background

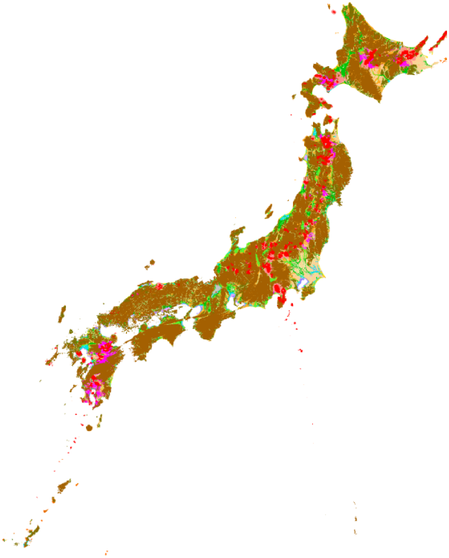


(c) Object

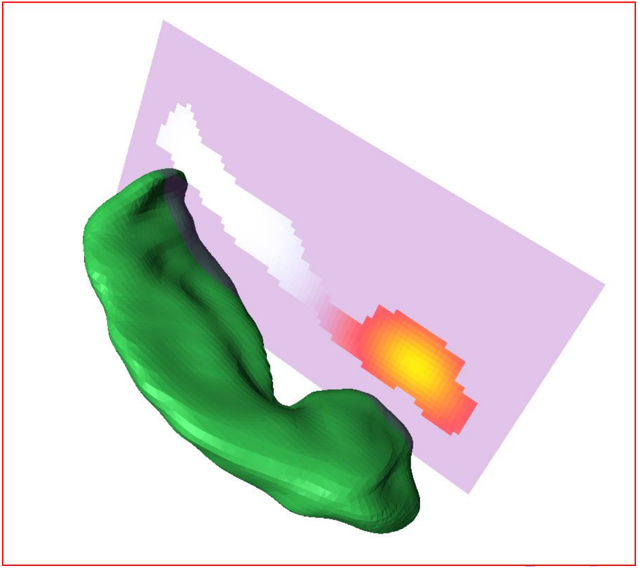


(d) Anomalies

Data Analysis on a Complicated Domain



3D Hippocampus Shape Analysis



Outline

Motivations: Why Graphs?

- More and more data are collected in a distributed and irregular manner; they are not organized such as familiar digital signals and images sampled on regular lattices. Examples include:
  - Data from sensor networks
  - Data from social networks, webpages, ...
  - Data from biological networks
  - ...
- It is quite important to analyze:
  - Topology of graphs/networks (e.g., how nodes are connected, etc.)
  - Data measured on nodes (e.g., a node = a sensor, then what is an edge?)

Motivations: Why Graphs?

- More and more data are collected in a distributed and irregular manner; they are not organized such as familiar digital signals and images sampled on regular lattices. Examples include:
  - Data from sensor networks
  - Data from social networks, webpages, ...
  - Data from biological networks
  - ...
- It is quite important to analyze:
  - Topology of graphs/networks (e.g., how nodes are connected, etc.)
  - Data measured on nodes (e.g., a node = a sensor, then what is an edge?)

Motivations: Why Graphs?

- More and more data are collected in a distributed and irregular manner; they are not organized such as familiar digital signals and images sampled on regular lattices. Examples include:
  - Data from sensor networks
  - Data from social networks, webpages, ...
  - Data from biological networks
  - ...
- It is quite important to analyze:
  - Topology of graphs/networks (e.g., how nodes are connected, etc.)
  - Data measured on nodes (e.g., a node = a sensor, then what is an edge?)

Motivations: Why Graphs?

- More and more data are collected in a distributed and irregular manner; they are not organized such as familiar digital signals and images sampled on regular lattices. Examples include:
  - Data from sensor networks
  - Data from social networks, webpages, . . .
  - Data from biological networks
  - . . .
- It is quite important to analyze:
  - Topology of graphs/networks (e.g., how nodes are connected, etc.)
  - Data measured on nodes (e.g., a node = a sensor, then what is an edge?)

Motivations: Why Graphs?

- More and more data are collected in a distributed and irregular manner; they are not organized such as familiar digital signals and images sampled on regular lattices. Examples include:
  - Data from sensor networks
  - Data from social networks, webpages, . . .
  - Data from biological networks
  - . . .
- It is quite important to analyze:
  - Topology of graphs/networks (e.g., how nodes are connected, etc.)
  - Data measured on nodes (e.g., a node = a sensor, then what is an edge?)

Motivations: Why Graphs?

- More and more data are collected in a distributed and irregular manner; they are not organized such as familiar digital signals and images sampled on regular lattices. Examples include:
  - Data from sensor networks
  - Data from social networks, webpages, ...
  - Data from biological networks
  - ...
- It is quite important to analyze:
  - Topology of graphs/networks (e.g., how nodes are connected, etc.)
  - Data measured on nodes (e.g., a node = a sensor, then what is an edge?)

Motivations: Why Graphs?

- More and more data are collected in a distributed and irregular manner; they are not organized such as familiar digital signals and images sampled on regular lattices. Examples include:
    - Data from sensor networks
    - Data from social networks, webpages, . . .
    - Data from biological networks
    - . . .
- It is quite important to analyze:
    - Topology of graphs/networks (e.g., how nodes are connected, etc.)
    - Data measured on nodes (e.g., a node = a sensor, then what is an edge?)

Motivations: Why Graphs?

- More and more data are collected in a distributed and irregular manner; they are not organized such as familiar digital signals and images sampled on regular lattices. Examples include:
    - Data from sensor networks
    - Data from social networks, webpages, ...
    - Data from biological networks
    - ...
- It is quite important to analyze:
    - Topology of graphs/networks (e.g., how nodes are connected, etc.)
    - Data measured on nodes (e.g., a node = a sensor, then what is an edge?)

Motivations: Why Graphs?

- Fourier analysis/synthesis and wavelet analysis/synthesis have been 'crown jewels' for data sampled on the regular lattices.
- Hence, we need to lift such tools for unorganized and irregularly-sampled datasets including those represented by graphs and networks.
- Moreover, constructing a graph from a usual signal or image and analyzing it can also be very useful! E.g., Nonlocal means image denoising of Buades-Coll-Morel.

Motivations: Why Graphs?

- Fourier analysis/synthesis and wavelet analysis/synthesis have been 'crown jewels' for data sampled on the regular lattices.
- Hence, we need to lift such tools for unorganized and irregularly-sampled datasets including those represented by graphs and networks.
- Moreover, constructing a graph from a usual signal or image and analyzing it can also be very useful! E.g., Nonlocal means image denoising of Buades-Coll-Morel.

Motivations: Why Graphs?

- Fourier analysis/synthesis and wavelet analysis/synthesis have been 'crown jewels' for data sampled on the regular lattices.
- Hence, we need to lift such tools for unorganized and irregularly-sampled datasets including those represented by graphs and networks.
- Moreover, constructing a graph from a usual signal or image and analyzing it can also be very useful! E.g., Nonlocal means image denoising of Buades-Coll-Morel.

An Example of Sensor Networks

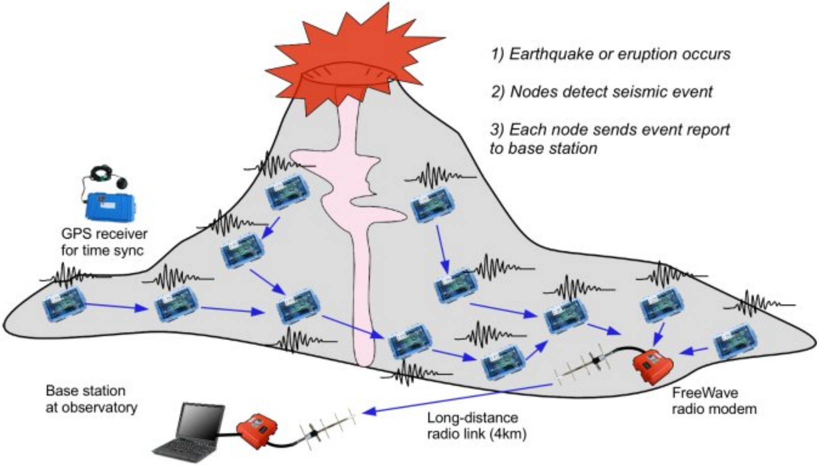


Figure: Volcano monitoring sensor network architecture of Harvard Sensor Networks Lab

An Example of Social Networks

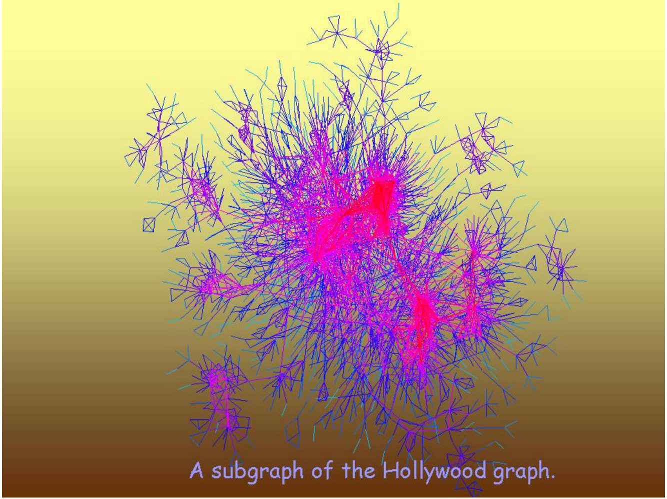


Figure: Through the courtesy of Prof. Fan Chung, UC San Diego

An Example of Biological Networks

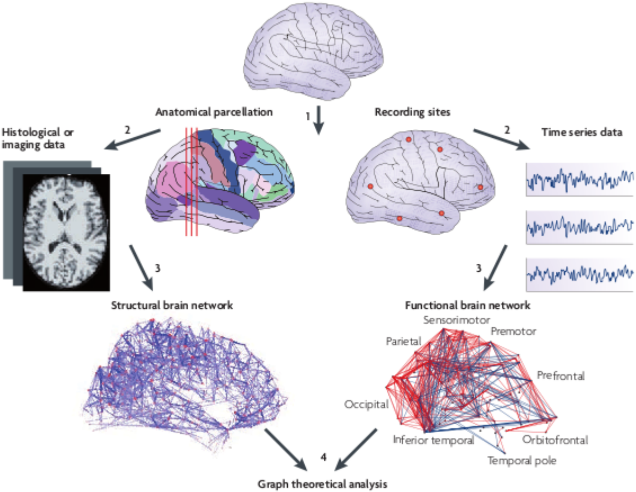
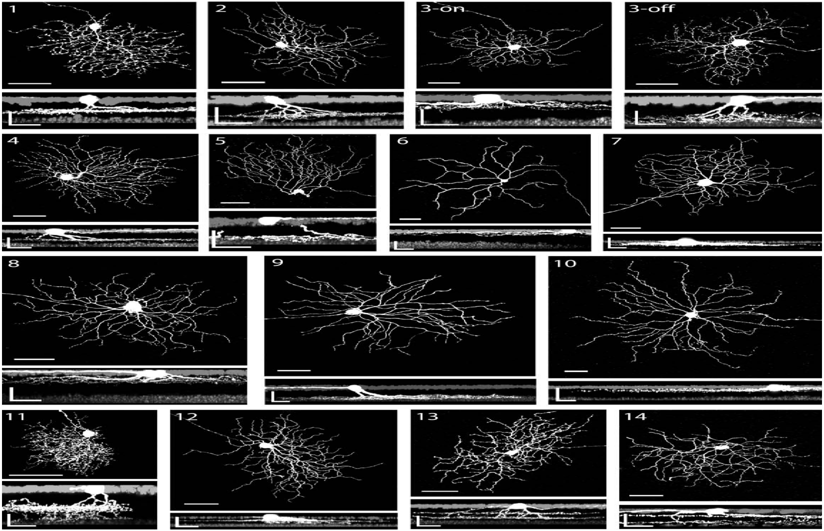
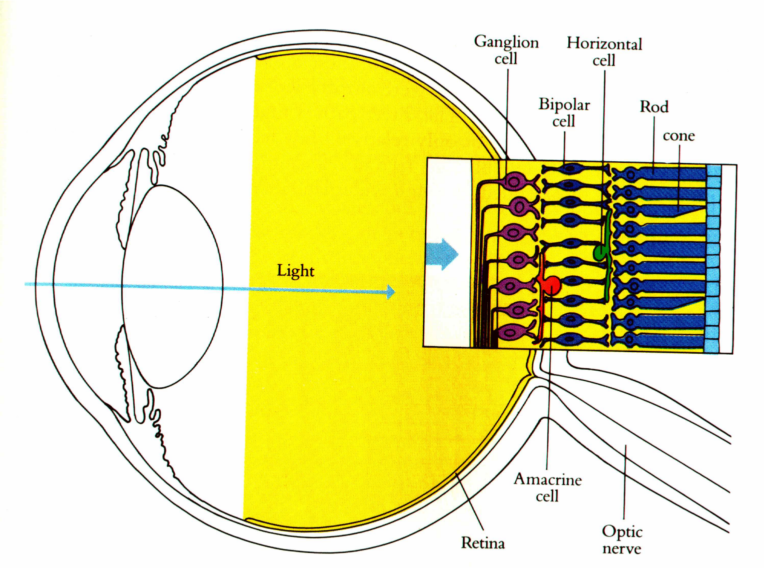


Figure: From E. Bullmore and O. Sporns, *Nature Reviews Neuroscience*, vol. 10, pp.186–198, Mar. 2009.

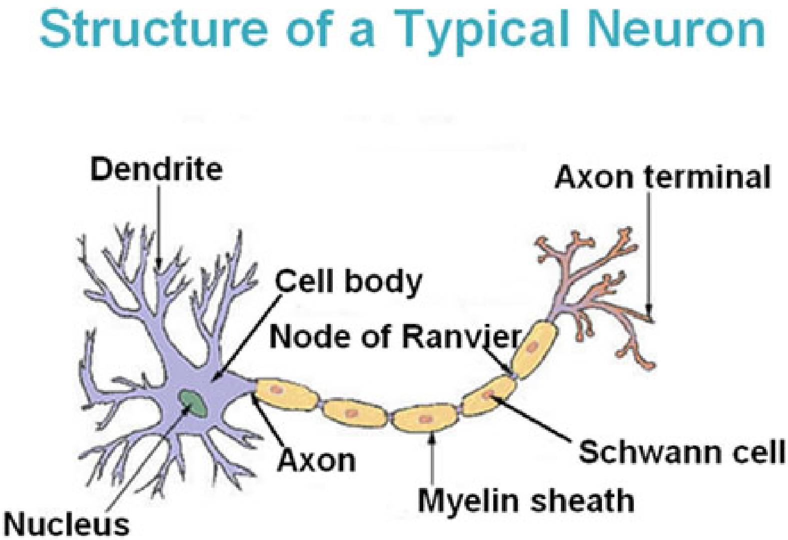
Another Biological Example: Retinal Ganglion Cells



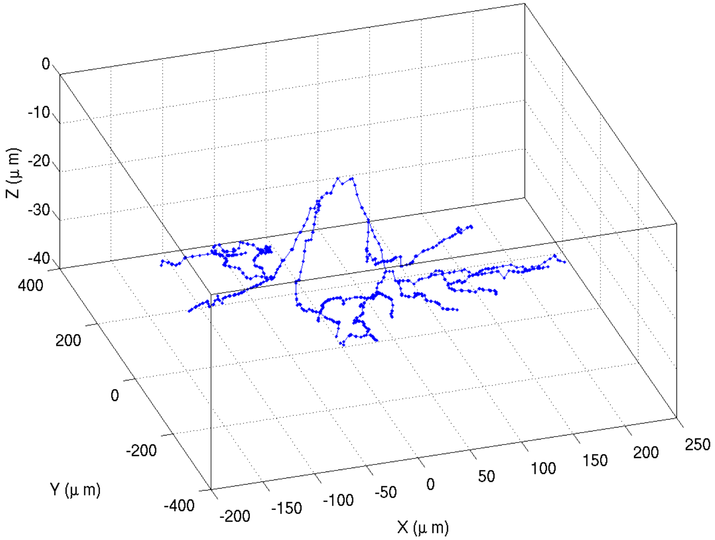
Retinal Ganglion Cells (D. Hubel: *Eye, Brain, & Vision*, '95)



A Typical Neuron (from Wikipedia)



Mouse's RGC as a Graph



Enter Laplacian Eigenfunctions!

- On either irregular Euclidean domains or graphs, appropriately defined *Laplacian eigenfunctions* play an important role for data analysis.
- Let us first consider an irregular (i.e., general shape) Euclidean domain $\Omega \subset \mathbb{R}^d$.
- Let $\mathscr{L} := -\Delta = -\left(\frac{\partial^2}{\partial {x_1}^2} + \cdots + \frac{\partial^2}{\partial {x_d}^2}\right)$.
- The Laplacian eigenvalue problem is defined as:

$$\mathscr{L} u = -\Delta u = \lambda u \quad \text{in } \Omega,$$

together with some appropriate boundary condition (BC).
- Most common (homogeneous) BCs are:
  - *Dirichlet:* $u = 0$ on $\partial\Omega$;
  - *Neumann:* $\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$;
  - *Robin (or impedance):* $au + b\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$, $a \neq 0 \neq b$.

Enter Laplacian Eigenfunctions!

- On either irregular Euclidean domains or graphs, appropriately defined *Laplacian eigenfunctions* play an important role for data analysis.
- Let us first consider an irregular (i.e., general shape) Euclidean domain $\Omega \subset \mathbb{R}^d$.
- Let $\mathscr{L} := -\Delta = -\left(\frac{\partial^2}{\partial {x_1}^2} + \cdots + \frac{\partial^2}{\partial {x_d}^2}\right)$.
- The Laplacian eigenvalue problem is defined as:

$$\mathscr{L} u = -\Delta u = \lambda u \quad \text{in } \Omega,$$

together with some appropriate boundary condition (BC).
- Most common (homogeneous) BCs are:
    - *Dirichlet:* $u = 0$ on $\partial\Omega$;
    - *Neumann:* $\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$;
    - *Robin (or impedance):* $au + b\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$, $a \neq 0 \neq b$.

Enter Laplacian Eigenfunctions!

- On either irregular Euclidean domains or graphs, appropriately defined *Laplacian eigenfunctions* play an important role for data analysis.
- Let us first consider an irregular (i.e., general shape) Euclidean domain $\Omega \subset \mathbb{R}^d$.
- Let $\mathscr{L} := -\Delta = -\left(\frac{\partial^2}{\partial {x_1}^2} + \cdots + \frac{\partial^2}{\partial {x_d}^2}\right)$.
- The Laplacian eigenvalue problem is defined as:

$$\mathscr{L} u = -\Delta u = \lambda u \quad \text{in } \Omega,$$

  together with some *appropriate* boundary condition (BC).
- Most common (homogeneous) BCs are:
  - *Dirichlet:* $u = 0$ on $\partial\Omega$;
  - *Neumann:* $\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$;
  - *Robin (or impedance):* $au + b\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$, $a \neq 0 \neq b$.

Enter Laplacian Eigenfunctions!

- On either irregular Euclidean domains or graphs, appropriately defined *Laplacian eigenfunctions* play an important role for data analysis.
- Let us first consider an irregular (i.e., general shape) Euclidean domain $\Omega \subset \mathbb{R}^d$.
- Let $\mathscr{L} := -\Delta = -\left(\frac{\partial^2}{\partial {x_1}^2} + \cdots + \frac{\partial^2}{\partial {x_d}^2}\right)$.
- The Laplacian eigenvalue problem is defined as:

$$\mathscr{L} u = -\Delta u = \lambda u \quad \text{in } \Omega,$$

together with some appropriate boundary condition (BC).

- Most common (homogeneous) BCs are:
  - *Dirichlet:* $u = 0$ on $\partial\Omega$;
  - *Neumann:* $\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$;
  - *Robin (or impedance):* $au + b\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$, $a \neq 0 \neq b$.

Enter Laplacian Eigenfunctions!

- On either irregular Euclidean domains or graphs, appropriately defined *Laplacian eigenfunctions* play an important role for data analysis.
- Let us first consider an irregular (i.e., general shape) Euclidean domain $\Omega \subset \mathbb{R}^d$.
- Let $\mathscr{L} := -\Delta = -\left(\frac{\partial^2}{\partial {x_1}^2} + \cdots + \frac{\partial^2}{\partial {x_d}^2}\right)$.
- The Laplacian eigenvalue problem is defined as:

$$\mathscr{L} u = -\Delta u = \lambda u \quad \text{in } \Omega,$$

  together with some appropriate boundary condition (BC).
- Most common (homogeneous) BCs are:
  - *Dirichlet:* $u = 0$ on $\partial\Omega$;
  - *Neumann:* $\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$;
  - *Robin (or impedance):* $au + b\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$, $a \neq 0 \neq b$.

Enter Laplacian Eigenfunctions!

- On either irregular Euclidean domains or graphs, appropriately defined *Laplacian eigenfunctions* play an important role for data analysis.
- Let us first consider an irregular (i.e., general shape) Euclidean domain $\Omega \subset \mathbb{R}^d$.
- Let $\mathscr{L} := -\Delta = -\left(\frac{\partial^2}{\partial {x_1}^2} + \cdots + \frac{\partial^2}{\partial {x_d}^2}\right)$.
- The Laplacian eigenvalue problem is defined as:

$$\mathscr{L} u = -\Delta u = \lambda u \quad \text{in } \Omega,$$

  together with some appropriate boundary condition (BC).
- Most common (homogeneous) BCs are:
  - *Dirichlet:* $u = 0$ on $\partial\Omega$;
  - *Neumann:* $\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$;
  - *Robin (or impedance):* $au + b\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$, $a \neq 0 \neq b$.

Enter Laplacian Eigenfunctions!

- On either irregular Euclidean domains or graphs, appropriately defined *Laplacian eigenfunctions* play an important role for data analysis.
- Let us first consider an irregular (i.e., general shape) Euclidean domain $\Omega \subset \mathbb{R}^d$.
- Let $\mathscr{L} := -\Delta = -\left(\frac{\partial^2}{\partial {x_1}^2} + \cdots + \frac{\partial^2}{\partial {x_d}^2}\right)$.
- The Laplacian eigenvalue problem is defined as:

$$\mathscr{L} u = -\Delta u = \lambda u \quad \text{in } \Omega,$$

together with some appropriate boundary condition (BC).

- Most common (homogeneous) BCs are:
  - *Dirichlet:* $u = 0$ on $\partial\Omega$;
  - *Neumann:* $\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$;
  - *Robin (or impedance):* $au + b\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$, $a \neq 0 \neq b$.

Enter Laplacian Eigenfunctions!

- On either irregular Euclidean domains or graphs, appropriately defined *Laplacian eigenfunctions* play an important role for data analysis.
- Let us first consider an irregular (i.e., general shape) Euclidean domain $\Omega \subset \mathbb{R}^d$.
- Let $\mathscr{L} := -\Delta = -\left(\frac{\partial^2}{\partial {x_1}^2} + \cdots + \frac{\partial^2}{\partial {x_d}^2}\right)$.
- The Laplacian eigenvalue problem is defined as:

$$\mathscr{L} u = -\Delta u = \lambda u \quad \text{in } \Omega,$$

  together with some appropriate boundary condition (BC).
- Most common (homogeneous) BCs are:
  - *Dirichlet:* $u = 0$ on $\partial\Omega$;
  - *Neumann:* $\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$;
  - *Robin (or impedance):* $au + b\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$, $a \neq 0 \neq b$.

Enter Laplacian Eigenfunctions . . .

- The nontrivial solution $u = \varphi$ of such a *boundary value problem* (BVP) is called the Laplacian eigenfunction corresponding to the eigenvalue λ.
- Via Green's 1st identity, the Dirichlet BC leads to: $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k \to \infty$.
- On the other hand, the Neumann BC leads to: $0 = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k \to \infty$.
- In the case of the Robin BC, some eigenvalues may be even *negative*.

Enter Laplacian Eigenfunctions . . .

- The nontrivial solution $u = \varphi$ of such a *boundary value problem* (BVP) is called the Laplacian eigenfunction corresponding to the eigenvalue λ.
- Via Green's 1st identity, the Dirichlet BC leads to: $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k \to \infty$.
- On the other hand, the Neumann BC leads to: $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k \to \infty$.
- In the case of the Robin BC, some eigenvalues may be even *negative*.

Enter Laplacian Eigenfunctions . . .

- The nontrivial solution $u = \varphi$ of such a *boundary value problem* (BVP) is called the Laplacian eigenfunction corresponding to the eigenvalue λ.
- Via Green's 1st identity, the Dirichlet BC leads to: $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k \to \infty$.
- On the other hand, the Neumann BC leads to: $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k \to \infty$.
- In the case of the Robin BC, some eigenvalues may be even *negative*.

Enter Laplacian Eigenfunctions . . .

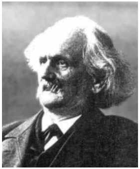
- The nontrivial solution $u = \varphi$ of such a *boundary value problem* (BVP) is called the Laplacian eigenfunction corresponding to the eigenvalue λ.
- Via Green's 1st identity, the Dirichlet BC leads to:
  $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k \to \infty$.
- On the other hand, the Neumann BC leads to:
  $0 = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k \to \infty$.
- In the case of the Robin BC, some eigenvalues may be even *negative*.

Enter Laplacian Eigenfunctions . . .

- The nontrivial solution $u = \varphi$ of such a *boundary value problem* (BVP) is called the Laplacian eigenfunction corresponding to the eigenvalue λ.
- Via Green's 1st identity, the Dirichlet BC leads to:
  $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k \to \infty$.
- On the other hand, the Neumann BC leads to:
  $0 = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k \to \infty$.
- In the case of the Robin BC, some eigenvalues may be even *negative*.

(a) P.-S. Laplace (1749–1827)

(b) J.P.G.L. Dirichlet (1805–1859)



(c) Carl Neumann (1832–1925)

(d) Gustave Robin (1855–1897)

Laplacian Eigenfunctions . . . Why?

- Why not analyze (and synthesize) an object of interest defined or measured on an irregular domain Ω using genuine basis functions tailored to the domain instead of the basis functions developed for rectangles, tori, balls, etc.?
- After all, *sines* (and *cosines*) are the eigenfunctions of the Laplacian on a *rectangular* domain (e.g., an interval in 1D) with Dirichlet (and Neumann) boundary condition.
- *Spherical harmonics, Bessel functions, and Prolate Spheroidal Wave Functions*, are part of the eigenfunctions of the Laplacian (via separation of variables) for the *spherical, cylindrical, and spheroidal* domains, respectively.
- Laplacian eigenfunctions (LEs) allow us to perform spectral analysis of data measured at more general domains or even on graphs and networks $\Longrightarrow$ Generalization of Fourier analysis!

Laplacian Eigenfunctions . . . Why?

- Why not analyze (and synthesize) an object of interest defined or measured on an irregular domain Ω using genuine basis functions tailored to the domain instead of the basis functions developed for rectangles, tori, balls, etc.?
- After all, *sines* (and *cosines*) are the eigenfunctions of the Laplacian on a *rectangular* domain (e.g., an interval in 1D) with Dirichlet (and Neumann) boundary condition.
- *Spherical harmonics, Bessel functions, and Prolate Spheroidal Wave Functions*, are part of the eigenfunctions of the Laplacian (via separation of variables) for the *spherical, cylindrical, and spheroidal* domains, respectively.
- Laplacian eigenfunctions (LEs) allow us to perform spectral analysis of data measured at more general domains or even on graphs and networks $\Longrightarrow$ Generalization of Fourier analysis!

Laplacian Eigenfunctions . . . Why?

- Why not analyze (and synthesize) an object of interest defined or measured on an irregular domain Ω using genuine basis functions tailored to the domain instead of the basis functions developed for rectangles, tori, balls, etc.?
- After all, *sines* (and *cosines*) are the eigenfunctions of the Laplacian on a *rectangular* domain (e.g., an interval in 1D) with Dirichlet (and Neumann) boundary condition.
- *Spherical harmonics, Bessel functions, and Prolate Spheroidal Wave Functions*, are part of the eigenfunctions of the Laplacian (via separation of variables) for the *spherical, cylindrical, and spheroidal* domains, respectively.
- Laplacian eigenfunctions (LEs) allow us to perform spectral analysis of data measured at more general domains or even on graphs and networks $\Longrightarrow$ Generalization of Fourier analysis!

Laplacian Eigenfunctions . . . Why?

- Why not analyze (and synthesize) an object of interest defined or measured on an irregular domain Ω using genuine basis functions tailored to the domain instead of the basis functions developed for rectangles, tori, balls, etc.?
- After all, *sines* (and *cosines*) are the eigenfunctions of the Laplacian on a *rectangular* domain (e.g., an interval in 1D) with Dirichlet (and Neumann) boundary condition.
- *Spherical harmonics, Bessel functions, and Prolate Spheroidal Wave Functions*, are part of the eigenfunctions of the Laplacian (via separation of variables) for the *spherical, cylindrical, and spheroidal* domains, respectively.
- Laplacian eigenfunctions (LEs) allow us to perform spectral analysis of data measured at more general domains or even on graphs and networks $\Longrightarrow$ Generalization of Fourier analysis!

Laplacian Eigenfunctions . . . Why?

- LEs have more physical meaning (i.e., vibration modes, heat conduction, . . .) than other popular basis functions such as *wavelets and wavelet packets*.
- LEs may particularly be useful for inverse problems and imaging: Suppose the domain shape Ω is fixed yet the material contents inside that domain, say $u(x)$, $x \in \Omega$, change over time, i.e., $u(x,t)$, $x \in \Omega$, $t \in [0,T]$. Suppose one want to detect whether there is any change in the material contents in Ω over time, i.e., estimate $u_t(x,t)$ via imaging.
- LEs may also be necessary for many shape optimization problems: e.g., among all possible 2D shapes having unit area, what is the shape that minimizes its *fifth* smallest Dirichlet-Laplacian eigenvalues?

Laplacian Eigenfunctions . . . Why?

- LEs have more physical meaning (i.e., vibration modes, heat conduction, . . .) than other popular basis functions such as *wavelets and wavelet packets*.
- LEs may particularly be useful for inverse problems and imaging: Suppose the domain shape Ω is fixed yet the material contents inside that domain, say $u(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$, change over time, i.e., $u(\boldsymbol{x}, t)$, $\boldsymbol{x} \in \Omega$, $t \in [0, T]$. Suppose one want to detect whether there is any change in the material contents in Ω over time, i.e., estimate $u_t(\boldsymbol{x}, t)$ via imaging.
- LEs may also be necessary for many shape optimization problems: e.g., among all possible 2D shapes having unit area, what is the shape that minimizes its *fifth* smallest Dirichlet-Laplacian eigenvalues?

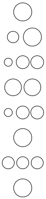
Laplacian Eigenfunctions . . . Why?

- LEs have more physical meaning (i.e., vibration modes, heat conduction, . . .) than other popular basis functions such as *wavelets and wavelet packets*.
- LEs may particularly be useful for inverse problems and imaging: Suppose the domain shape Ω is fixed yet the material contents inside that domain, say $u(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$, change over time, i.e., $u(\boldsymbol{x}, t)$, $\boldsymbol{x} \in \Omega$, $t \in [0, T]$. Suppose one want to detect whether there is any change in the material contents in Ω over time, i.e., estimate $u_t(\boldsymbol{x}, t)$ via imaging.
- LEs may also be necessary for many shape optimization problems: e.g., among all possible 2D shapes having unit area, what is the shape that minimizes its *fifth* smallest Dirichlet-Laplacian eigenvalues?

Shape Optimization (Courtesy of B. Osting)

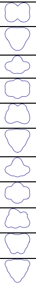
Computational results for single eigenvalues

Oudet (2004)

No	Optimal union of discs		Computed shapes	
3		46.125		46.125
4		64.293		64.293
5		82.462		**78.47**
6		92.250		**88.96**
7		110.42		**107.47**
8		127.88		**119.9**
9		138.37		**133.52**
10		154.62		**143.45**

- The level set method is used to represent the domains
- Relaxed formulation used to compute eigenvalues
- The k-th eigenvalue of the minimizer is multiple

Antunes + Freitas (2012)

i	Ω	multiplicity	λ_i^*	Oudet's result
5		2	**78.20**	78.47
6		3	**88.52**	88.96
7		3	**106.14**	107.47
8		3	**118.90**	119.9
9		3	**132.68**	133.52
10		4	**142.72**	143.45
11		4	**159.39**	-
12		4	**172.85**	-
13		4	**186.97**	-
14		4	**198.96**	-
15		5	**209.63**	-

- Eigenvalues computed via meshless method
- Domains parameterized using Fourier coefficients
- $k = 13$ minimizer is not symmetric

Laplacian Eigenfunctions . . . Some Facts

- Analysis of $\mathscr{L}$ is difficult due to its unboundedness, etc.
- Much better to analyze its inverse, i.e., the Green's operator because it is compact and self-adjoint.
- Thus $\mathscr{L}^{-1}$ has discrete spectra (i.e., a countable number of eigenvalues with finite multiplicity) except 0 spectrum.
- $\mathscr{L}$ has a complete orthonormal basis of $L^2(\Omega)$, and this allows us to do eigenfunction expansion in $L^2(\Omega)$.

Laplacian Eigenfunctions . . . Some Facts

- Analysis of $\mathscr{L}$ is difficult due to its unboundedness, etc.
- Much better to analyze its inverse, i.e., the Green's operator because it is compact and self-adjoint.
- Thus $\mathscr{L}^{-1}$ has discrete spectra (i.e., a countable number of eigenvalues with finite multiplicity) except 0 spectrum.
- $\mathscr{L}$ has a complete orthonormal basis of $L^2(\Omega)$, and this allows us to do eigenfunction expansion in $L^2(\Omega)$.

Laplacian Eigenfunctions . . . Some Facts

- Analysis of $\mathscr{L}$ is difficult due to its unboundedness, etc.
- Much better to analyze its inverse, i.e., the Green's operator because it is compact and self-adjoint.
- Thus $\mathscr{L}^{-1}$ has discrete spectra (i.e., a countable number of eigenvalues with finite multiplicity) except 0 spectrum.
- $\mathscr{L}$ has a complete orthonormal basis of $L^2(\Omega)$, and this allows us to do eigenfunction expansion in $L^2(\Omega)$.

Laplacian Eigenfunctions . . . Some Facts

- Analysis of $\mathscr{L}$ is difficult due to its unboundedness, etc.
- Much better to analyze its inverse, i.e., the Green's operator because it is compact and self-adjoint.
- Thus $\mathscr{L}^{-1}$ has discrete spectra (i.e., a countable number of eigenvalues with finite multiplicity) except 0 spectrum.
- $\mathscr{L}$ has a complete orthonormal basis of $L^2(\Omega)$, and this allows us to do eigenfunction expansion in $L^2(\Omega)$.

Laplacian Eigenfunctions . . . Difficulties

- The key difficulty is to compute such eigenfunctions; directly solving the Helmholtz equation (or eigenvalue problem) on a general domain is tough.
- Unfortunately, computing the Green's function for a general Ω satisfying the usual boundary condition (i.e., Dirichlet, Neumann, Robin) is also very difficult.

Laplacian Eigenfunctions . . . Difficulties

- The key difficulty is to compute such eigenfunctions; directly solving the Helmholtz equation (or eigenvalue problem) on a general domain is tough.
- Unfortunately, computing the Green's function for a general Ω satisfying the usual boundary condition (i.e., Dirichlet, Neumann, Robin) is also very difficult.

Outline

Outline

Laplacian Eigenfunctions in 1D — The Wave Equation

Around mid 18 C, d'Alembert, Euler, D. Bernoulli examined and created the theory behind vibrations of a 1D string.

- Consider a perfectly elastic and flexible string of length ℓ.
- $\rho(x)$: a mass density; $T(x)$: the tension of the string at $x \in [0, \ell]$.
- If $u(x,t)$ is the vertical displacement of the string at location $x \in [0, \ell]$ and time $t \geq 0$, then the string vibrates according to the 1D wave equation (a.k.a. the string equation): $\rho(x)\dfrac{\partial^2 u}{\partial t^2} = \dfrac{\partial}{\partial x}\left(T(x)\dfrac{\partial u}{\partial x}\right)$

Laplacian Eigenfunctions in 1D — The Wave Equation

Around mid 18 C, d'Alembert, Euler, D. Bernoulli examined and created the theory behind vibrations of a 1D string.

- Consider a perfectly elastic and flexible string of length ℓ.
- $\rho(x)$: a mass density; $T(x)$: the tension of the string at $x \in [0, \ell]$.
- If $u(x,t)$ is the vertical displacement of the string at location $x \in [0, \ell]$ and time $t \geq 0$, then the string vibrates according to the 1D wave equation (a.k.a. the string equation): $\rho(x) \frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x}\left(T(x) \frac{\partial u}{\partial x} \right)$

Laplacian Eigenfunctions in 1D — The Wave Equation

Around mid 18 C, d'Alembert, Euler, D. Bernoulli examined and created the theory behind vibrations of a 1D string.

- Consider a perfectly elastic and flexible string of length ℓ.
- $\rho(x)$: a mass density; $T(x)$: the tension of the string at $x \in [0, \ell]$.
- If $u(x,t)$ is the vertical displacement of the string at location $x \in [0, \ell]$ and time $t \geq 0$, then the string vibrates according to the 1D wave equation (a.k.a. the string equation): $\rho(x)\dfrac{\partial^2 u}{\partial t^2} = \dfrac{\partial}{\partial x}\left(T(x)\dfrac{\partial u}{\partial x}\right)$

Laplacian Eigenfunctions in 1D — The Wave Equation

Around mid 18 C, d'Alembert, Euler, D. Bernoulli examined and created the theory behind vibrations of a 1D string.

- Consider a perfectly elastic and flexible string of length ℓ.
- $\rho(x)$: a mass density; $T(x)$: the tension of the string at $x \in [0, \ell]$.
- If $u(x,t)$ is the vertical displacement of the string at location $x \in [0, \ell]$ and time $t \geq 0$, then the string vibrates according to the 1D wave equation (a.k.a. the string equation): $\rho(x)\dfrac{\partial^2 u}{\partial t^2} = \dfrac{\partial}{\partial x}\left(T(x)\dfrac{\partial u}{\partial x}\right)$

(a) Jean d'Alembert (1717–1783)

(b) Leonhard Euler (1707–1783)

(c) Daniel Bernoulli (1700–1782)

Importance of the Boundary and Initial Conditions

- From now on, for simplicity, we assume the uniform density and constant tension, i.e., $\rho(x) \equiv \rho$, $T(x) \equiv T$.
- Under this assumption, the above wave equation simplifies to:

$$u_{tt} = c^2 u_{xx} \quad c \equiv \sqrt{T/\rho}.$$

- The 1D wave equation above has infinitely many solutions.
- Need to specify a boundary condition (BC) and an initial condition (IC) to obtain the desired solution.
- One possibility: both ends of the string are held fixed all the time $\Longrightarrow$ the Dirichlet BC: $u(0,t) = u(\ell,t) = 0$, $\forall t \geq 0$.
- As for the IC, let $u(x,0) = f(x)$ (initial position); $u_t(x,0) = g(x)$ (initial velocity), $\forall x \in [0,\ell]$. What we have then is:

$$\begin{cases} u_{tt} = c^2 u_{xx} & \text{for } x \in (0,\ell) \text{ and } t > 0; \\ u(0,t) = u(\ell,t) = 0 & \text{for } t \geq 0; \\ u(x,0) = f(x), \quad u_t(x,0) = g(x) & \text{for } x \in [0,\ell]. \end{cases} \tag{1}$$

Importance of the Boundary and Initial Conditions

- From now on, for simplicity, we assume the uniform density and constant tension, i.e., $\rho(x) \equiv \rho$, $T(x) \equiv T$.
- Under this assumption, the above wave equation simplifies to:

$$u_{tt} = c^2 u_{xx} \quad c \equiv \sqrt{T/\rho}.$$

- The 1D wave equation above has infinitely many solutions.
- Need to specify a boundary condition (BC) and an initial condition (IC) to obtain the desired solution.
- One possibility: both ends of the string are held fixed all the time $\Longrightarrow$ the Dirichlet BC: $u(0,t) = u(\ell,t) = 0$, $\forall t \geq 0$.
- As for the IC, let $u(x,0) = f(x)$ (initial position); $u_t(x,0) = g(x)$ (initial velocity), $\forall x \in [0,\ell]$. What we have then is:

$$\begin{cases} u_{tt} = c^2 u_{xx} & \text{for } x \in (0,\ell) \text{ and } t > 0; \\ u(0,t) = u(\ell,t) = 0 & \text{for } t \geq 0; \\ u(x,0) = f(x), \quad u_t(x,0) = g(x) & \text{for } x \in [0,\ell]. \end{cases} \qquad (1)$$

Importance of the Boundary and Initial Conditions

- From now on, for simplicity, we assume the uniform density and constant tension, i.e., $\rho(x) \equiv \rho$, $T(x) \equiv T$.
- Under this assumption, the above wave equation simplifies to:

$$u_{tt} = c^2 u_{xx} \quad c \equiv \sqrt{T/\rho}.$$

- The 1D wave equation above has infinitely many solutions.
- Need to specify a boundary condition (BC) and an initial condition (IC) to obtain the desired solution.
- One possibility: both ends of the string are held fixed all the time $\Longrightarrow$ the Dirichlet BC: $u(0,t) = u(\ell,t) = 0$, $\forall t \geq 0$.
- As for the IC, let $u(x,0) = f(x)$ (initial position); $u_t(x,0) = g(x)$ (initial velocity), $\forall x \in [0,\ell]$. What we have then is:

$$\begin{cases} u_{tt} = c^2 u_{xx} & \text{for } x \in (0,\ell) \text{ and } t > 0; \\ u(0,t) = u(\ell,t) = 0 & \text{for } t \geq 0; \\ u(x,0) = f(x), \quad u_t(x,0) = g(x) & \text{for } x \in [0,\ell]. \end{cases} \quad (1)$$

Importance of the Boundary and Initial Conditions

- From now on, for simplicity, we assume the uniform density and constant tension, i.e., $\rho(x) \equiv \rho$, $T(x) \equiv T$.
- Under this assumption, the above wave equation simplifies to:

$$u_{tt} = c^2 u_{xx} \quad c \equiv \sqrt{T/\rho}.$$

- The 1D wave equation above has infinitely many solutions.
- Need to specify a boundary condition (BC) and an initial condition (IC) to obtain the desired solution.
- One possibility: both ends of the string are held fixed all the time $\Longrightarrow$ the Dirichlet BC: $u(0,t) = u(\ell,t) = 0$, $\forall t \geq 0$.
- As for the IC, let $u(x,0) = f(x)$ (initial position); $u_t(x,0) = g(x)$ (initial velocity), $\forall x \in [0,\ell]$. What we have then is:

$$\begin{cases} u_{tt} = c^2 u_{xx} & \text{for } x \in (0,\ell) \text{ and } t > 0; \\ u(0,t) = u(\ell,t) = 0 & \text{for } t \geq 0; \\ u(x,0) = f(x), \quad u_t(x,0) = g(x) & \text{for } x \in [0,\ell]. \end{cases} \tag{1}$$

Importance of the Boundary and Initial Conditions

- From now on, for simplicity, we assume the uniform density and constant tension, i.e., $\rho(x) \equiv \rho$, $T(x) \equiv T$.
- Under this assumption, the above wave equation simplifies to:

$$u_{tt} = c^2 u_{xx} \quad c \equiv \sqrt{T/\rho}.$$

- The 1D wave equation above has infinitely many solutions.
- Need to specify a boundary condition (BC) and an initial condition (IC) to obtain the desired solution.
- One possibility: both ends of the string are held fixed all the time $\Longrightarrow$ the Dirichlet BC: $u(0,t) = u(\ell,t) = 0$, $\forall t \geq 0$.
- As for the IC, let $u(x,0) = f(x)$ (initial position); $u_t(x,0) = g(x)$ (initial velocity), $\forall x \in [0,\ell]$. What we have then is:

$$\begin{cases} u_{tt} = c^2 u_{xx} & \text{for } x \in (0,\ell) \text{ and } t > 0; \\ u(0,t) = u(\ell,t) = 0 & \text{for } t \geq 0; \\ u(x,0) = f(x), \quad u_t(x,0) = g(x) & \text{for } x \in [0,\ell]. \end{cases} \tag{1}$$

Importance of the Boundary and Initial Conditions

- From now on, for simplicity, we assume the uniform density and constant tension, i.e., $\rho(x) \equiv \rho$, $T(x) \equiv T$.
- Under this assumption, the above wave equation simplifies to:

$$u_{tt} = c^2 u_{xx} \quad c \equiv \sqrt{T/\rho}.$$

- The 1D wave equation above has infinitely many solutions.
- Need to specify a boundary condition (BC) and an initial condition (IC) to obtain the desired solution.
- One possibility: both ends of the string are held fixed all the time $\Longrightarrow$ the Dirichlet BC: $u(0,t) = u(\ell,t) = 0$, $\forall t \geq 0$.
- As for the IC, let $u(x,0) = f(x)$ (initial position); $u_t(x,0) = g(x)$ (initial velocity), $\forall x \in [0,\ell]$. What we have then is:

$$\begin{cases} u_{tt} = c^2 u_{xx} & \text{for } x \in (0,\ell) \text{ and } t > 0; \\ u(0,t) = u(\ell,t) = 0 & \text{for } t \geq 0; \\ u(x,0) = f(x), \quad u_t(x,0) = g(x) & \text{for } x \in [0,\ell]. \end{cases} \tag{1}$$

Behavior of the String $u(x,t)$

- Use the method of separation of variables to seek a nontrivial solution of the form: $u(x,t) = X(x)T(t)$.
- Plugging $X(x)T(t)$ into the (1), we get:

$$XT'' = c^2X''T \Longrightarrow \frac{X''}{X} = \frac{T''}{c^2T} = k,$$

where k must be a *constant*.

- This leads to the following ODEs:

$$X'' - kX = 0 \quad \text{with } X(0) = X(\ell) = 0, \tag{2}$$

$$T'' - c^2kT = 0 \tag{3}$$

- The characteristic equation of (2), i.e., $r^2 - k = 0$, must be analyzed carefully.

Behavior of the String $u(x,t)$

- Use the method of separation of variables to seek a nontrivial solution of the form: $u(x,t) = X(x)T(t)$.
- Plugging $X(x)T(t)$ into the (1), we get:

$$XT'' = c^2 X'' T \Longrightarrow \frac{X''}{X} = \frac{T''}{c^2 T} = k,$$

where k must be a *constant*.

- This leads to the following ODEs:

$$X'' - kX = 0 \quad \text{with } X(0) = X(\ell) = 0, \tag{2}$$

$$T'' - c^2 kT = 0 \tag{3}$$

- The characteristic equation of (2), i.e., $r^2 - k = 0$, must be analyzed carefully.

Behavior of the String $u(x, t)$

- Use the method of separation of variables to seek a nontrivial solution of the form: $u(x,t) = X(x)T(t)$.
- Plugging $X(x)T(t)$ into the (1), we get:

$$XT'' = c^2 X'' T \Longrightarrow \frac{X''}{X} = \frac{T''}{c^2 T} = k,$$

where k must be a *constant*.

- This leads to the following ODEs:

$$X'' - kX = 0 \quad \text{with } X(0) = X(\ell) = 0, \tag{2}$$

$$T'' - c^2 kT = 0 \tag{3}$$

- The characteristic equation of (2), i.e., $r^2 - k = 0$, must be analyzed carefully.

Behavior of the String $u(x,t)$

- Use the method of separation of variables to seek a nontrivial solution of the form: $u(x,t) = X(x)T(t)$.
- Plugging $X(x)T(t)$ into the (1), we get:

$$XT'' = c^2 X'' T \Longrightarrow \frac{X''}{X} = \frac{T''}{c^2 T} = k,$$

where k must be a *constant*.
- This leads to the following ODEs:

$$X'' - kX = 0 \quad \text{with } X(0) = X(\ell) = 0, \tag{2}$$

$$T'' - c^2 kT = 0 \tag{3}$$

- The characteristic equation of (2), i.e., $r^2 - k = 0$, must be analyzed carefully.

Solving ODEs

Case I: $k > 0 \Longrightarrow r = \pm\sqrt{k}$; hence

$$X(x) = Ae^{\sqrt{k}x} + Be^{-\sqrt{k}x} \quad \text{or} \quad A\cosh(\sqrt{k}x) + B\sinh(\sqrt{k}x).$$

Applying the BC $X(0) = X(\ell) = 0$ yields $A = B = 0$, thus the case of $k > 0$ is *not feasible*.

Case II: $k = 0 \Longrightarrow X'' = 0 \Longrightarrow X(x) = Ax + B$, which again leads to $X(x) \equiv 0$.

Case III: $k < 0$. Set $k = -\xi^2$ and $\xi > 0$. Then the characteristic equation becomes $r^2 + \xi^2 = 0$, i.e., $r = \pm i\xi$. Therefore we get

$$X(x) = A\cos(\xi x) + B\sin(\xi x)$$

By the BC $X(0) = X(\ell) = 0$, we get:

$$\begin{cases} X(0) = 0 & \Longrightarrow \quad A = 0 \\ X(\ell) = B\sin(\xi\ell) = 0 & \Longrightarrow \quad \xi = \frac{n\pi}{\ell}, \quad \forall n \in \mathbb{N} \end{cases}$$

Note $n = 0$ leads to $X(x) \equiv 0$ in this case, so it should not be included.

Solving ODEs

Case I: $k > 0 \Longrightarrow r = \pm\sqrt{k}$; hence

$$X(x) = Ae^{\sqrt{k}x} + Be^{-\sqrt{k}x} \quad \text{or} \quad A\cosh(\sqrt{k}x) + B\sinh(\sqrt{k}x).$$

Applying the BC $X(0) = X(\ell) = 0$ yields $A = B = 0$, thus the case of $k > 0$ is *not feasible*.

Case II: $k = 0 \Longrightarrow X'' = 0 \Longrightarrow X(x) = Ax + B$, which again leads to $X(x) \equiv 0$.

Case III: $k < 0$. Set $k = -\xi^2$ and $\xi > 0$. Then the characteristic equation becomes $r^2 + \xi^2 = 0$, i.e., $r = \pm i\xi$. Therefore we get

$$X(x) = A\cos(\xi x) + B\sin(\xi x)$$

By the BC $X(0) = X(\ell) = 0$, we get:

$$\begin{cases} X(0) = 0 & \Longrightarrow \quad A = 0 \\ X(\ell) = B\sin(\xi\ell) = 0 & \Longrightarrow \quad \xi = \frac{n\pi}{\ell}, \quad \forall n \in \mathbb{N} \end{cases}$$

Note $n = 0$ leads to $X(x) \equiv 0$ in this case, so it should not be included.

Solving ODEs

Case I: $k > 0 \Longrightarrow r = \pm\sqrt{k}$; hence

$$X(x) = Ae^{\sqrt{k}x} + Be^{-\sqrt{k}x} \quad \text{or} \quad A\cosh(\sqrt{k}x) + B\sinh(\sqrt{k}x).$$

Applying the BC $X(0) = X(\ell) = 0$ yields $A = B = 0$, thus the case of $k > 0$ is *not feasible*.

Case II: $k = 0 \Longrightarrow X'' = 0 \Longrightarrow X(x) = Ax + B$, which again leads to $X(x) \equiv 0$.

Case III: $k < 0$. Set $k = -\xi^2$ and $\xi > 0$. Then the characteristic equation becomes $r^2 + \xi^2 = 0$, i.e., $r = \pm i\xi$. Therefore we get

$$X(x) = A\cos(\xi x) + B\sin(\xi x)$$

By the BC $X(0) = X(\ell) = 0$, we get:

$$\begin{cases} X(0) = 0 & \Longrightarrow \quad A = 0 \\ X(\ell) = B\sin(\xi\ell) = 0 & \Longrightarrow \quad \xi = \frac{n\pi}{\ell}, \quad \forall n \in \mathbb{N} \end{cases}$$

Note $n = 0$ leads to $X(x) \equiv 0$ in this case, so it should not be included.

Forming the Solution

- Hence we have $X(x) = B\sin(\frac{n\pi}{\ell}x)$, and for convenience, by setting $B = \sqrt{2/\ell}$, let us define

$$X_n(x) = \varphi_n(x) := \sqrt{\frac{2}{\ell}}\sin\left(\frac{n\pi}{\ell}x\right),$$

so that $\|\varphi_n\|_{L^2[0,\ell]} = 1$. Note that $\{\varphi_n\}_{n\in\mathbb{N}}$ form an orthonormal basis for $L^2[0,\ell]$.

- Similarly, by $T'' = -\xi^2 c^2 T$ we obtain the family of solutions

$$T_n(t) = a_n \cos\left(\frac{n\pi c}{\ell}t\right) + b_n \sin\left(\frac{n\pi c}{\ell}t\right).$$

- Now, for each $n \in \mathbb{N}$, the function

$$u_n(x,t) = T_n(t)\cdot\varphi_n(x) = \left\{a_n \cos\left(\frac{n\pi c}{\ell}t\right) + b_n \sin\left(\frac{n\pi c}{\ell}t\right)\right\}\sqrt{\frac{2}{\ell}}\sin\left(\frac{n\pi}{\ell}x\right)$$

satisfies (1).

Forming the Solution

- Hence we have $X(x) = B\sin(\frac{n\pi}{\ell}x)$, and for convenience, by setting $B = \sqrt{2/\ell}$, let us define

$$X_n(x) = \varphi_n(x) := \sqrt{\frac{2}{\ell}}\sin\left(\frac{n\pi}{\ell}x\right),$$

so that $\|\varphi_n\|_{L^2[0,\ell]} = 1$. Note that $\{\varphi_n\}_{n\in\mathbb{N}}$ form an orthonormal basis for $L^2[0,\ell]$.

- Similarly, by $T'' = -\xi^2c^2T$ we obtain the family of solutions

$$T_n(t) = a_n\cos\left(\frac{n\pi c}{\ell}t\right) + b_n\sin\left(\frac{n\pi c}{\ell}t\right).$$

- Now, for each $n \in \mathbb{N}$, the function

$$u_n(x,t) = T_n(t)\cdot\varphi_n(x) = \left\{a_n\cos\left(\frac{n\pi c}{\ell}t\right) + b_n\sin\left(\frac{n\pi c}{\ell}t\right)\right\}\sqrt{\frac{2}{\ell}}\sin\left(\frac{n\pi}{\ell}x\right)$$

satisfies (1).

Forming the Solution

- Hence we have $X(x) = B\sin(\frac{n\pi}{\ell}x)$, and for convenience, by setting $B = \sqrt{2/\ell}$, let us define

$$X_n(x) = \varphi_n(x) := \sqrt{\frac{2}{\ell}}\sin\left(\frac{n\pi}{\ell}x\right),$$

  so that $\|\varphi_n\|_{L^2[0,\ell]} = 1$. Note that $\{\varphi_n\}_{n\in\mathbb{N}}$ form an orthonormal basis for $L^2[0,\ell]$.
- Similarly, by $T'' = -\xi^2 c^2 T$ we obtain the family of solutions

$$T_n(t) = a_n\cos\left(\frac{n\pi c}{\ell}t\right) + b_n\sin\left(\frac{n\pi c}{\ell}t\right).$$

- Now, for each $n \in \mathbb{N}$, the function

$$u_n(x,t) = T_n(t)\cdot\varphi_n(x) = \left\{a_n\cos\left(\frac{n\pi c}{\ell}t\right) + b_n\sin\left(\frac{n\pi c}{\ell}t\right)\right\}\sqrt{\frac{2}{\ell}}\sin\left(\frac{n\pi}{\ell}x\right)$$

  satisfies (1).

Forming the Solution . . .

- Hence, by the *Superposition Principle*,

$$u(x,t) = \sum_{n=1}^{\infty} u_n(x,t) = \sum_{n=1}^{\infty} \left\{ a_n \cos\left(\frac{n\pi c}{\ell} t\right) + b_n \sin\left(\frac{n\pi c}{\ell} t\right) \right\} \varphi_n(x) \quad (4)$$

is a general solution with yet undetermined coefficients a_n and b_n.

- Next, we specify the coefficients a_n and b_n by matching (4) with the ICs in (1). Thus we get

$$u(x,0) = f(x) = \sum_{n=1}^{\infty} a_n \sqrt{\frac{2}{\ell}} \sin\left(\frac{n\pi}{\ell} x\right) = \sum_{n=1}^{\infty} a_n \varphi_n(x)$$

Then

$$a_n = \langle f, \varphi_n \rangle = \sqrt{\frac{2}{\ell}} \int_0^{\ell} f(x) \sin\left(\frac{n\pi}{\ell} x\right) \mathrm{d}x,$$

which is a Fourier sine series expansion of f.

Forming the Solution . . .

- Hence, by the *Superposition Principle*,

$$u(x,t) = \sum_{n=1}^{\infty} u_n(x,t) = \sum_{n=1}^{\infty} \left\{ a_n \cos\left(\frac{n\pi c}{\ell} t\right) + b_n \sin\left(\frac{n\pi c}{\ell} t\right) \right\} \varphi_n(x) \quad (4)$$

  is a general solution with yet undetermined coefficients a_n and b_n.

- Next, we specify the coefficients a_n and b_n by matching (4) with the ICs in (1). Thus we get

$$u(x,0) = f(x) = \sum_{n=1}^{\infty} a_n \sqrt{\frac{2}{\ell}} \sin\left(\frac{n\pi}{\ell} x\right) = \sum_{n=1}^{\infty} a_n \varphi_n(x)$$

  Then

$$a_n = \langle f, \varphi_n \rangle = \sqrt{\frac{2}{\ell}} \int_0^{\ell} f(x) \sin\left(\frac{n\pi}{\ell} x\right) \mathrm{d}x,$$

  which is a Fourier sine series expansion of f.

Forming the Solution ...

- Similarly, $u_t(x,0) = g(x) = \sum_{n=1}^{\infty} \frac{n\pi c}{\ell} b_n \sqrt{\frac{2}{\ell}} \sin\left(\frac{n\pi}{\ell}x\right)$.
- Note that $\frac{n\pi c}{\ell} b_n = \langle g, \varphi_n \rangle \Longrightarrow b_n = \frac{\ell}{n\pi c}\langle g, \varphi_n \rangle$.
- Finally, we obtain the particular solution:

$$u(x,t) = \sum_{n=1}^{\infty} \left\{ \langle f, \varphi_n \rangle \cos\left(\frac{n\pi c}{\ell} t\right) + \frac{\ell}{n\pi c} \langle g, \varphi_n \rangle \sin\left(\frac{n\pi c}{\ell} t\right) \right\} \varphi_n(x),$$

which satisfies (1) completely including both BC & IC.

Figure: Jean Baptiste Joseph Fourier (1768–1830)

Forming the Solution . . .

- Similarly, $u_t(x,0) = g(x) = \sum_{n=1}^{\infty} \frac{n\pi c}{\ell} b_n \sqrt{\frac{2}{\ell}} \sin\left(\frac{n\pi}{\ell} x\right)$.
- Note that $\frac{n\pi c}{\ell} b_n = \langle g, \varphi_n \rangle \Longrightarrow b_n = \frac{\ell}{n\pi c} \langle g, \varphi_n \rangle$.
- Finally, we obtain the particular solution:

$$u(x,t) = \sum_{n=1}^{\infty} \left\{ \langle f, \varphi_n \rangle \cos\left(\frac{n\pi c}{\ell} t\right) + \frac{\ell}{n\pi c} \langle g, \varphi_n \rangle \sin\left(\frac{n\pi c}{\ell} t\right) \right\} \varphi_n(x),$$

which satisfies (1) completely including both BC & IC.

Figure: Jean Baptiste Joseph Fourier (1768–1830)

Forming the Solution ...

- Similarly, $u_t(x,0) = g(x) = \sum_{n=1}^{\infty} \frac{n\pi c}{\ell} b_n \sqrt{\frac{2}{\ell}} \sin\left(\frac{n\pi}{\ell} x\right)$.
- Note that $\frac{n\pi c}{\ell} b_n = \langle g, \varphi_n \rangle \Longrightarrow b_n = \frac{\ell}{n\pi c} \langle g, \varphi_n \rangle$.
- Finally, we obtain the particular solution:

$$u(x,t) = \sum_{n=1}^{\infty} \left\{ \langle f, \varphi_n \rangle \cos\left(\frac{n\pi c}{\ell} t\right) + \frac{\ell}{n\pi c} \langle g, \varphi_n \rangle \sin\left(\frac{n\pi c}{\ell} t\right) \right\} \varphi_n(x),$$

which satisfies (1) completely including both BC & IC.

Figure: Jean Baptiste Joseph Fourier (1768–1830)

Remarks

- Need to check if our solution makes sense physically. Notice that

$$c^2 = \frac{T}{\rho} \Longrightarrow \text{the sound frequency} = \frac{n\pi}{\ell}\sqrt{\frac{T}{\rho}}.$$

- Hence, ℓ is short, T is high, and ρ is small (thin), then such a string generates a high frequency tone.
- On the other hand, if ℓ is long, T is low, and ρ is large (thick), then it generates a low frequency tone.
- Note that the Neumann BC imposes

$$u_x(0,t) = u_x(\ell,t) = 0 \quad \forall t > 0.$$

This leads to the Fourier cosine series expansions of f and g. Note that the Neumann problem allows the solution $u_0(x,t) = a_0 = \text{const}$.

Remarks

- Need to check if our solution makes sense physically. Notice that

$$c^2 = \frac{T}{\rho} \Longrightarrow \text{ the sound frequency } = \frac{n\pi}{\ell}\sqrt{\frac{T}{\rho}}.$$

- Hence, ℓ is short, T is high, and ρ is small (thin), then such a string generates a high frequency tone.
- On the other hand, if ℓ is long, T is low, and ρ is large (thick), then it generates a low frequency tone.
- Note that the Neumann BC imposes

$$u_x(0,t) = u_x(\ell,t) = 0 \quad \forall t > 0.$$

This leads to the Fourier cosine series expansions of f and g. Note that the Neumann problem allows the solution $u_0(x,t) = a_0 = \text{const.}$

Remarks

- Need to check if our solution makes sense physically. Notice that

$$c^2 = \frac{T}{\rho} \Longrightarrow \text{ the sound frequency } = \frac{n\pi}{\ell}\sqrt{\frac{T}{\rho}}.$$

- Hence, ℓ is short, T is high, and ρ is small (thin), then such a string generates a high frequency tone.
- On the other hand, if ℓ is long, T is low, and ρ is large (thick), then it generates a low frequency tone.
- Note that the Neumann BC imposes

$$u_x(0,t) = u_x(\ell,t) = 0 \quad \forall t > 0.$$

This leads to the Fourier cosine series expansions of f and g. Note that the Neumann problem allows the solution $u_0(x,t) = a_0 = \text{const}$.

Remarks

- Need to check if our solution makes sense physically. Notice that

$$c^2 = \frac{T}{\rho} \Longrightarrow \text{the sound frequency} = \frac{n\pi}{\ell}\sqrt{\frac{T}{\rho}}.$$

- Hence, ℓ is short, T is high, and ρ is small (thin), then such a string generates a high frequency tone.
- On the other hand, if ℓ is long, T is low, and ρ is large (thick), then it generates a low frequency tone.
- Note that the Neumann BC imposes

$$u_x(0,t) = u_x(\ell,t) = 0 \quad \forall t > 0.$$

This leads to the Fourier cosine series expansions of f and g. Note that the Neumann problem allows the solution $u_0(x,t) = a_0 = \text{const}$.

Remarks . . .

- Through the separation of variables for finding a solution to the 1D string equation with BC & IC (1), we arrive at the system

$$-X'' = \xi^2 X \quad \text{with } X(0) = X(\ell) = 0. \tag{5}$$

- Notice that (5) is a 1D version of the Dirichlet-Laplacian eigenvalue problem with $\Omega = (0, \ell)$.
- More importantly, we obtained two objects, namely:

$$\text{Eigenvalues: } \lambda_n^D = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N};$$

$$\text{Eigenfunctions: } \varphi_n^D(x) = \sqrt{\frac{2}{\ell}} \sin\left(\sqrt{\lambda_n^D}\, x\right) \quad n \in \mathbb{N}.$$

- In the case of the Neumann-Laplacian, we got

$$\text{Eigenvalues: } \lambda_n^N = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N}_0 := \{0\} \cup \mathbb{N};$$

$$\text{Eigenfunctions: } \varphi_n^N(x) = \sqrt{\frac{2}{\ell}} \cos\left(\sqrt{\lambda_n^N}\, x\right) \quad n \in \mathbb{N}_0.$$

Remarks . . .

- Through the separation of variables for finding a solution to the 1D string equation with BC & IC (1), we arrive at the system

$$-X'' = \xi^2 X \quad \text{with } X(0) = X(\ell) = 0. \tag{5}$$

- Notice that (5) is a 1D version of the Dirichlet-Laplacian eigenvalue problem with $\Omega = (0, \ell)$.
- More importantly, we obtained two objects, namely:

$$\text{Eigenvalues: } \lambda_n^D = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N};$$

$$\text{Eigenfunctions: } \varphi_n^D(x) = \sqrt{\frac{2}{\ell}} \sin\left(\sqrt{\lambda_n^D}\, x\right) \quad n \in \mathbb{N}.$$

- In the case of the Neumann-Laplacian, we got

$$\text{Eigenvalues: } \lambda_n^N = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N}_0 := \{0\} \cup \mathbb{N};$$

$$\text{Eigenfunctions: } \varphi_n^N(x) = \sqrt{\frac{2}{\ell}} \cos\left(\sqrt{\lambda_n^N}\, x\right) \quad n \in \mathbb{N}_0.$$

Remarks . . .

- Through the separation of variables for finding a solution to the 1D string equation with BC & IC (1), we arrive at the system

$$-X'' = \xi^2 X \quad \text{with } X(0) = X(\ell) = 0. \tag{5}$$

- Notice that (5) is a 1D version of the Dirichlet-Laplacian eigenvalue problem with $\Omega = (0, \ell)$.
- More importantly, we obtained two objects, namely:

Eigenvalues: $\lambda_n^D = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N};$

Eigenfunctions: $\varphi_n^D(x) = \sqrt{\frac{2}{\ell}} \sin\left(\sqrt{\lambda_n^D}\, x\right) \quad n \in \mathbb{N}.$

- In the case of the Neumann-Laplacian, we got

Eigenvalues: $\lambda_n^N = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N}_0 := \{0\} \cup \mathbb{N};$

Eigenfunctions: $\varphi_n^N(x) = \sqrt{\frac{2}{\ell}} \cos\left(\sqrt{\lambda_n^N}\, x\right) \quad n \in \mathbb{N}_0.$

Remarks . . .

- Through the separation of variables for finding a solution to the 1D string equation with BC & IC (1), we arrive at the system

$$-X'' = \xi^2 X \quad \text{with } X(0) = X(\ell) = 0. \tag{5}$$

- Notice that (5) is a 1D version of the Dirichlet-Laplacian eigenvalue problem with $\Omega = (0, \ell)$.
- More importantly, we obtained two objects, namely:

Eigenvalues: $\lambda_n^D = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N};$

Eigenfunctions: $\varphi_n^D(x) = \sqrt{\frac{2}{\ell}} \sin\left(\sqrt{\lambda_n^D}\, x\right) \quad n \in \mathbb{N}.$

- In the case of the Neumann-Laplacian, we got

Eigenvalues: $\lambda_n^N = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N}_0 := \{0\} \cup \mathbb{N};$

Eigenfunctions: $\varphi_n^N(x) = \sqrt{\frac{2}{\ell}} \cos\left(\sqrt{\lambda_n^N}\, x\right) \quad n \in \mathbb{N}_0.$

Remarks . . .

- Through the separation of variables for finding a solution to the 1D string equation with BC & IC (1), we arrive at the system

$$-X'' = \xi^2 X \quad \text{with } X(0) = X(\ell) = 0. \tag{5}$$

- Notice that (5) is a 1D version of the Dirichlet-Laplacian eigenvalue problem with $\Omega = (0, \ell)$.
- More importantly, we obtained two objects, namely:

  Eigenvalues: $\lambda_n^D = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N};$

  Eigenfunctions: $\varphi_n^D(x) = \sqrt{\frac{2}{\ell}} \sin\left(\sqrt{\lambda_n^D}\, x\right) \quad n \in \mathbb{N}.$

- In the case of the Neumann-Laplacian, we got

  Eigenvalues: $\lambda_n^N = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N}_0 := \{0\} \cup \mathbb{N};$

  Eigenfunctions: $\varphi_n^N(x) = \sqrt{\frac{2}{\ell}} \cos\left(\sqrt{\lambda_n^N}\, x\right) \quad n \in \mathbb{N}_0$

Remarks . . .

- Through the separation of variables for finding a solution to the 1D string equation with BC & IC (1), we arrive at the system

$$-X'' = \xi^2 X \quad \text{with } X(0) = X(\ell) = 0. \tag{5}$$

- Notice that (5) is a 1D version of the Dirichlet-Laplacian eigenvalue problem with $\Omega = (0, \ell)$.
- More importantly, we obtained two objects, namely:

Eigenvalues: $\lambda_n^D = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N};$

Eigenfunctions: $\varphi_n^D(x) = \sqrt{\frac{2}{\ell}} \sin\left(\sqrt{\lambda_n^D}\, x\right) \quad n \in \mathbb{N}.$

- In the case of the Neumann-Laplacian, we got

Eigenvalues: $\lambda_n^N = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N}_0 := \{0\} \cup \mathbb{N};$

Eigenfunctions: $\varphi_n^N(x) = \sqrt{\frac{2}{\ell}} \cos\left(\sqrt{\lambda_n^N}\, x\right) \quad n \in \mathbb{N}_0.$

Remarks . . .

- Through the separation of variables for finding a solution to the 1D string equation with BC & IC (1), we arrive at the system

$$-X'' = \xi^2 X \quad \text{with } X(0) = X(\ell) = 0. \tag{5}$$

- Notice that (5) is a 1D version of the Dirichlet-Laplacian eigenvalue problem with $\Omega = (0, \ell)$.
- More importantly, we obtained two objects, namely:

Eigenvalues: $\lambda_n^D = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N};$

Eigenfunctions: $\varphi_n^D(x) = \sqrt{\frac{2}{\ell}} \sin\left(\sqrt{\lambda_n^D}\, x\right) \quad n \in \mathbb{N}.$

- In the case of the Neumann-Laplacian, we got

Eigenvalues: $\lambda_n^N = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N}_0 := \{0\} \cup \mathbb{N};$

Eigenfunctions: $\varphi_n^N(x) = \sqrt{\frac{2}{\ell}} \cos\left(\sqrt{\lambda_n^N}\, x\right) \quad n \in \mathbb{N}_0.$

Remarks . . .

- Through the separation of variables for finding a solution to the 1D string equation with BC & IC (1), we arrive at the system

$$-X'' = \xi^2 X \quad \text{with } X(0) = X(\ell) = 0. \tag{5}$$

- Notice that (5) is a 1D version of the Dirichlet-Laplacian eigenvalue problem with $\Omega = (0, \ell)$.
- More importantly, we obtained two objects, namely:

Eigenvalues: $\lambda_n^D = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N};$

Eigenfunctions: $\varphi_n^D(x) = \sqrt{\frac{2}{\ell}} \sin\left(\sqrt{\lambda_n^D}\, x\right) \quad n \in \mathbb{N}.$

- In the case of the Neumann-Laplacian, we got

Eigenvalues: $\lambda_n^N = \left(\frac{n\pi}{\ell}\right)^2 \quad n \in \mathbb{N}_0 := \{0\} \cup \mathbb{N};$

Eigenfunctions: $\varphi_n^N(x) = \sqrt{\frac{2}{\ell}} \cos\left(\sqrt{\lambda_n^N}\, x\right) \quad n \in \mathbb{N}_0.$

Remarks . . .

- We see that in either BCs, $\{\lambda_n\}_{n=1}^{\infty}$ contains *geometric information* of the domain $\Omega = (0, \ell)$.
- For instance, the size of the first eigenvalue, $\lambda_1 = (\pi/\ell)^2$ tells us the volume of Ω (i.e., the length ℓ of Ω in 1D).
- Under our assumption of constant tension and constant density,

$$\text{small } \lambda_1 \iff \text{long } \ell$$
$$\text{large } \lambda_1 \iff \text{short } \ell$$

- Furthermore, the set $\{\varphi_n\}_{n=1}^{\infty}$ forms an orthonormal basis for $L^2(\Omega)$, so the eigenfunctions allows us to analyze functions living on Ω.

Remarks . . .

- We see that in either BCs, $\{\lambda_n\}_{n=1}^{\infty}$ contains *geometric information* of the domain $\Omega = (0, \ell)$.
- For instance, the size of the first eigenvalue, $\lambda_1 = (\pi/\ell)^2$ tells us the volume of Ω (i.e., the length ℓ of Ω in 1D).
- Under our assumption of constant tension and constant density,

$$\text{small } \lambda_1 \iff \text{long } \ell$$
$$\text{large } \lambda_1 \iff \text{short } \ell$$

- Furthermore, the set $\{\varphi_n\}_{n=1}^{\infty}$ forms an orthonormal basis for $L^2(\Omega)$, so the eigenfunctions allows us to analyze functions living on Ω.

Remarks . . .

- We see that in either BCs, $\{\lambda_n\}_{n=1}^{\infty}$ contains *geometric information* of the domain $\Omega = (0, \ell)$.
- For instance, the size of the first eigenvalue, $\lambda_1 = (\pi/\ell)^2$ tells us the volume of Ω (i.e., the length ℓ of Ω in 1D).
- Under our assumption of constant tension and constant density,

$$\text{small } \lambda_1 \iff \text{long } \ell$$
$$\text{large } \lambda_1 \iff \text{short } \ell$$

- Furthermore, the set $\{\varphi_n\}_{n=1}^{\infty}$ forms an orthonormal basis for $L^2(\Omega)$, so the eigenfunctions allows us to analyze functions living on Ω.

Remarks . . .

- We see that in either BCs, $\{\lambda_n\}_{n=1}^{\infty}$ contains *geometric information* of the domain $\Omega = (0, \ell)$.
- For instance, the size of the first eigenvalue, $\lambda_1 = (\pi/\ell)^2$ tells us the volume of Ω (i.e., the length ℓ of Ω in 1D).
- Under our assumption of constant tension and constant density,

$$\text{small } \lambda_1 \iff \text{long } \ell$$
$$\text{large } \lambda_1 \iff \text{short } \ell$$

- Furthermore, the set $\{\varphi_n\}_{n=1}^{\infty}$ forms an orthonormal basis for $L^2(\Omega)$, so the eigenfunctions allows us to analyze functions living on Ω.

Outline

Spectral Geometry 101

- The Laplacian eigenfunctions defined on the domain Ω provides the orthonormal basis of $L^2(\Omega)$.
- The Laplacian eigenvalues encode geometric information of the domain $\Omega \Longrightarrow$ "Can we hear the shape of a drum?" (Mark Kac, 1966).
- Temporarily, consider the Laplacian eigenvalue problem on a planar domain $\Omega \in \mathbb{R}^2$ with the *Dirichlet* boundary condition:

$$\begin{cases} -\Delta u = \lambda u & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

- Let $0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_k \leq \cdots \to \infty$ be the sequence of eigenvalues of the above Dirichlet-Laplace eigenvalue problem.

Spectral Geometry 101

- The Laplacian eigenfunctions defined on the domain Ω provides the orthonormal basis of $L^2(\Omega)$.
- The Laplacian eigenvalues encode geometric information of the domain $\Omega \Longrightarrow$ "Can we hear the shape of a drum?" (Mark Kac, 1966).
- Temporarily, consider the Laplacian eigenvalue problem on a planar domain $\Omega \in \mathbb{R}^2$ with the *Dirichlet* boundary condition:

$$\begin{cases} -\Delta u = \lambda u & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

- Let $0 < \lambda_1 \le \lambda_2 \le \lambda_3 \le \cdots \le \lambda_k \le \cdots \to \infty$ be the sequence of eigenvalues of the above Dirichlet-Laplace eigenvalue problem.

Spectral Geometry 101

- The Laplacian eigenfunctions defined on the domain Ω provides the orthonormal basis of $L^2(\Omega)$.
- The Laplacian eigenvalues encode geometric information of the domain $\Omega \Longrightarrow$ "Can we hear the shape of a drum?" (Mark Kac, 1966).
- Temporarily, consider the Laplacian eigenvalue problem on a planar domain $\Omega \in \mathbb{R}^2$ with the *Dirichlet* boundary condition:

$$\begin{cases} -\Delta u = \lambda u & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

- Let $0 < \lambda_1 \le \lambda_2 \le \lambda_3 \le \cdots \le \lambda_k \le \cdots \to \infty$ be the sequence of eigenvalues of the above Dirichlet-Laplace eigenvalue problem.

Spectral Geometry 101

- The Laplacian eigenfunctions defined on the domain Ω provides the orthonormal basis of $L^2(\Omega)$.
- The Laplacian eigenvalues encode geometric information of the domain $\Omega \Longrightarrow$ "Can we hear the shape of a drum?" (Mark Kac, 1966).
- Temporarily, consider the Laplacian eigenvalue problem on a planar domain $\Omega \in \mathbb{R}^2$ with the *Dirichlet* boundary condition:

$$\begin{cases} -\Delta u = \lambda u & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

- Let $0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_k \leq \cdots \to \infty$ be the sequence of eigenvalues of the above Dirichlet-Laplace eigenvalue problem.

Spectral Geometry 101 . . .

Kac showed (based on the work of Weyl, Minakshisundaram-Pleijel):

$$\sum_{k=1}^{\infty} \mathrm{e}^{-\lambda_k t} = \frac{|\Omega|}{4\pi t} - \frac{|\partial\Omega|}{8\sqrt{\pi t}} + o(t^{-1/2}) \quad \text{as } t \downarrow 0.$$

Spectral Geometry 101 . . .

Kac showed (based on the work of Weyl, Minakshisundaram-Pleijel):

$$\sum_{k=1}^{\infty} e^{-\lambda_k t} = \frac{|\Omega|}{4\pi t} - \frac{|\partial\Omega|}{8\sqrt{\pi t}} + o(t^{-1/2}) \quad \text{as } t \downarrow 0.$$

(a) Hermann Weyl (1885–1955)

(b) S. Minakshisundaram (1913–1968)

(c) Åke Pleijel (1913–1989)

(d) Mark Kac (1914–1984)

Universal (or Payne-Pólya-Weinberger) Inequalities ($m \in \mathbb{N}$)

- $\lambda_{m+1} - \lambda_m \leq 2 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j$; $\lambda_{m+1} \leq 3 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j$; $\frac{\lambda_{m+1}}{\lambda_m} \leq 3.$
- $\sum_{j=1}^{m} \frac{\lambda_j}{\lambda_{m+1} - \lambda_j} \geq \frac{m}{2}$ (Hile-Protter).
- $\sum_{j=1}^{m} (\lambda_{m+1} - \lambda_j)^2 \leq 2 \sum_{j=1}^{m} \lambda_j (\lambda_{m+1} - \lambda_j)$ (Yang).

Universal (or Payne-Pólya-Weinberger) Inequalities ($m \in \mathbb{N}$)

- $\lambda_{m+1} - \lambda_m \le 2 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j$; $\quad \lambda_{m+1} \le 3 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j$; $\quad \frac{\lambda_{m+1}}{\lambda_m} \le 3$.
- $\sum_{j=1}^{m} \frac{\lambda_j}{\lambda_{m+1} - \lambda_j} \ge \frac{m}{2}$ (Hile-Protter).
- $\sum_{j=1}^{m} (\lambda_{m+1} - \lambda_j)^2 \le 2 \sum_{j=1}^{m} \lambda_j (\lambda_{m+1} - \lambda_j)$ (Yang).

Universal (or Payne-Pólya-Weinberger) Inequalities ($m \in \mathbb{N}$)

- $\lambda_{m+1} - \lambda_m \leq 2 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j\,; \quad \lambda_{m+1} \leq 3 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j\,; \quad \frac{\lambda_{m+1}}{\lambda_m} \leq 3.$
- $\sum_{j=1}^{m} \frac{\lambda_j}{\lambda_{m+1} - \lambda_j} \geq \frac{m}{2}$ (Hile-Protter).
- $\sum_{j=1}^{m} (\lambda_{m+1} - \lambda_j)^2 \leq 2 \sum_{j=1}^{m} \lambda_j (\lambda_{m+1} - \lambda_j)$ (Yang).

Universal (or Payne-Pólya-Weinberger) Inequalities ($m \in \mathbb{N}$)

- $\lambda_{m+1} - \lambda_m \le 2 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j$; $\quad \lambda_{m+1} \le 3 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j$; $\quad \frac{\lambda_{m+1}}{\lambda_m} \le 3.$
- $\sum_{j=1}^{m} \frac{\lambda_j}{\lambda_{m+1} - \lambda_j} \ge \frac{m}{2}$ (Hile-Protter).
- $\sum_{j=1}^{m} (\lambda_{m+1} - \lambda_j)^2 \le 2 \sum_{j=1}^{m} \lambda_j (\lambda_{m+1} - \lambda_j)$ (Yang).

(a) L. E. Payne (1923–2011)

(b) G. Pólya (1887–1985)

(c) H. Weinberger (1928–)

Isoperimetric Inequalities

- $\lambda_1 \geq \dfrac{\pi^2 j_{0,1}^2}{|\Omega|^2}$ (Rayleigh-Faber-Krahn)
- $\dfrac{\lambda_2}{\lambda_1} \leq \dfrac{j_{1,1}^2}{j_{0,1}^2} \approx 2.5387$ (Ashbaugh-Benguria)
- $j_{k,1}$ is the first zero of the Bessel function of order k, i.e., $J_k(j_{k,1}) = 0$. $j_{0,1} \approx 2.4048$, $j_{1,1} \approx 3.8317$, and $|\Omega|$ is the area of Ω. In both cases, the equality is attained iff Ω is a disk in $\mathbb{R}^2$.

Isoperimetric Inequalities

- $\lambda_1 \geq \dfrac{\pi^2 j_{0,1}^2}{|\Omega|^2}$ (Rayleigh-Faber-Krahn)
- $\dfrac{\lambda_2}{\lambda_1} \leq \dfrac{j_{1,1}^2}{j_{0,1}^2} \approx 2.5387$ (Ashbaugh-Benguria)
- $j_{k,1}$ is the first zero of the Bessel function of order k, i.e., $J_k(j_{k,1}) = 0$. $j_{0,1} \approx 2.4048$, $j_{1,1} \approx 3.8317$, and $|\Omega|$ is the area of Ω. In both cases, the equality is attained iff Ω is a disk in $\mathbb{R}^2$.

Isoperimetric Inequalities

- $\lambda_1 \geq \dfrac{\pi^2 j_{0,1}^2}{|\Omega|^2}$ (Rayleigh-Faber-Krahn)
- $\dfrac{\lambda_2}{\lambda_1} \leq \dfrac{j_{1,1}^2}{j_{0,1}^2} \approx 2.5387$ (Ashbaugh-Benguria)
- $j_{k,1}$ is the first zero of the Bessel function of order k, i.e., $J_k(j_{k,1}) = 0$. $j_{0,1} \approx 2.4048$, $j_{1,1} \approx 3.8317$, and $|\Omega|$ is the area of Ω. In both cases, the equality is attained iff Ω is a disk in $\mathbb{R}^2$.

Isoperimetric Inequalities

- $\lambda_1 \geq \dfrac{\pi^2 j_{0,1}^2}{|\Omega|^2}$ (Rayleigh-Faber-Krahn)
- $\dfrac{\lambda_2}{\lambda_1} \leq \dfrac{j_{1,1}^2}{j_{0,1}^2} \approx 2.5387$ (Ashbaugh-Benguria)
- $j_{k,1}$ is the first zero of the Bessel function of order k, i.e., $J_k(j_{k,1}) = 0$. $j_{0,1} \approx 2.4048$, $j_{1,1} \approx 3.8317$, and $|\Omega|$ is the area of Ω. In both cases, the equality is attained iff Ω is a disk in $\mathbb{R}^2$.

(a) Lord Rayleigh (1842–1919)

(b) Georg Faber (1877–1966)

(c) Edgar Krahn (1894–1961)

(d) Mark Ashbaugh (1953-)

(e) Rafael Benguria (1951-)

Remarks

Excellent references on these inequalities are:

- R. D. Benguria, H. Linde, & B. Loewe: "Isoperimetric inequalities for eigenvalues of the Laplacian and the Schrödinger operator," *Bull. Math. Sci.*, vol. 2, pp. 1–56, 2012.
- A. Henrot: *Extremum Problems for Eigenvalues of Elliptic Operators*, Birkhäuser Verlag, Basel, 2006.

Other Properties

- Domain monotonicity property: $\Omega_1 \subset \Omega_2 \Longrightarrow \lambda_k(\Omega_1) \geq \lambda_k(\Omega_2), \quad k \in \mathbb{N}$.
- Scaling property: $\lambda_k(\alpha\Omega) = \dfrac{\lambda_k(\Omega)}{\alpha^2}, \quad \alpha > 0, k \in \mathbb{N}$.

  This implies:

$$\frac{\lambda_k(\alpha\Omega)}{\lambda_m(\alpha\Omega)} = \frac{\lambda_k(\Omega)}{\lambda_m(\Omega)}, \quad k, m \in \mathbb{N}.$$

  $\Longrightarrow$ *the ratios of Laplacian eigenvalues are scale invariant*.
- Laplacian eigenvalues are translation and rotation invariant.
- Using these eigenvalues and eigenvalue ratios for shape recognition and classification has been quite popular recently as I will describe later.
- Some properties and inequalities listed above should hold not only for the Dirichlet Laplacian eigenvalues but also for our Laplacian eigenvalues. Note, however, that *the domain monotonicity does not hold for the Neumann Laplacian eigenvalues*.

Other Properties

- Domain monotonicity property: $\Omega_1 \subset \Omega_2 \Longrightarrow \lambda_k(\Omega_1) \geq \lambda_k(\Omega_2), \quad k \in \mathbb{N}$.
- Scaling property: $\lambda_k(\alpha\Omega) = \dfrac{\lambda_k(\Omega)}{\alpha^2}, \quad \alpha > 0,\, k \in \mathbb{N}$.

  This implies:

$$\frac{\lambda_k(\alpha\Omega)}{\lambda_m(\alpha\Omega)} = \frac{\lambda_k(\Omega)}{\lambda_m(\Omega)}, \quad k, m \in \mathbb{N}.$$

  $\Longrightarrow$ *the ratios of Laplacian eigenvalues are scale invariant*.
- Laplacian eigenvalues are translation and rotation invariant.
- Using these eigenvalues and eigenvalue ratios for shape recognition and classification has been quite popular recently as I will describe later.
- Some properties and inequalities listed above should hold not only for the Dirichlet Laplacian eigenvalues but also for our Laplacian eigenvalues. Note, however, that *the domain monotonicity does not hold for the Neumann Laplacian eigenvalues*.

Other Properties

- Domain monotonicity property: $\Omega_1 \subset \Omega_2 \Longrightarrow \lambda_k(\Omega_1) \geq \lambda_k(\Omega_2), \quad k \in \mathbb{N}.$
- Scaling property: $\lambda_k(\alpha\Omega) = \dfrac{\lambda_k(\Omega)}{\alpha^2}, \quad \alpha > 0, k \in \mathbb{N}.$

  This implies:

$$\frac{\lambda_k(\alpha\Omega)}{\lambda_m(\alpha\Omega)} = \frac{\lambda_k(\Omega)}{\lambda_m(\Omega)}, \quad k, m \in \mathbb{N}.$$

  $\Longrightarrow$ *the ratios of Laplacian eigenvalues are scale invariant*.
- Laplacian eigenvalues are translation and rotation invariant.
- Using these eigenvalues and eigenvalue ratios for shape recognition and classification has been quite popular recently as I will describe later.
- Some properties and inequalities listed above should hold not only for the Dirichlet Laplacian eigenvalues but also for our Laplacian eigenvalues. Note, however, that *the domain monotonicity does not hold for the Neumann Laplacian eigenvalues*.

Other Properties

- Domain monotonicity property: $\Omega_1 \subset \Omega_2 \Longrightarrow \lambda_k(\Omega_1) \geq \lambda_k(\Omega_2), \quad k \in \mathbb{N}$.
- Scaling property: $\lambda_k(\alpha\Omega) = \dfrac{\lambda_k(\Omega)}{\alpha^2}, \quad \alpha > 0,\, k \in \mathbb{N}$.

  This implies:
  $$\frac{\lambda_k(\alpha\Omega)}{\lambda_m(\alpha\Omega)} = \frac{\lambda_k(\Omega)}{\lambda_m(\Omega)}, \quad k, m \in \mathbb{N}.$$
  $\Longrightarrow$ *the ratios of Laplacian eigenvalues are scale invariant.*
- Laplacian eigenvalues are translation and rotation invariant.
- Using these eigenvalues and eigenvalue ratios for shape recognition and classification has been quite popular recently as I will describe later.
- Some properties and inequalities listed above should hold not only for the Dirichlet Laplacian eigenvalues but also for our Laplacian eigenvalues. Note, however, that *the domain monotonicity does not hold for the Neumann Laplacian eigenvalues.*

Other Properties

- Domain monotonicity property: $\Omega_1 \subset \Omega_2 \Longrightarrow \lambda_k(\Omega_1) \geq \lambda_k(\Omega_2), \quad k \in \mathbb{N}$.
- Scaling property: $\lambda_k(\alpha\Omega) = \dfrac{\lambda_k(\Omega)}{\alpha^2}, \quad \alpha > 0, k \in \mathbb{N}$.

  This implies:
  $$\frac{\lambda_k(\alpha\Omega)}{\lambda_m(\alpha\Omega)} = \frac{\lambda_k(\Omega)}{\lambda_m(\Omega)}, \quad k, m \in \mathbb{N}.$$
  $\Longrightarrow$ *the ratios of Laplacian eigenvalues are scale invariant*.
- Laplacian eigenvalues are translation and rotation invariant.
- Using these eigenvalues and eigenvalue ratios for shape recognition and classification has been quite popular recently as I will describe later.
- Some properties and inequalities listed above should hold not only for the Dirichlet Laplacian eigenvalues but also for our Laplacian eigenvalues. Note, however, that *the domain monotonicity does not hold for the Neumann Laplacian eigenvalues*.

An Counterexample to the Domain Monotonicity

Consider a $2D$ rectangle of sides a and b with $a > b$. Then, let $\Omega' := \{(x, y) \mid 0 < x < a,\ 0 < y < b\}$, and $\Omega \subset \Omega'$ be the inscribed thin rectangle of sides $\sqrt{\alpha^2 + \beta^2} \times \sqrt{(a-\alpha)^2 + (b-\beta)^2}$:

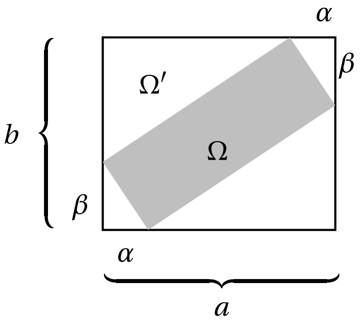


Figure: The Neumann BC generates an counterexample (From A. Henrot, 2006)

An Counterexample to the Domain Monotonicity ...

- Can easily compute the Neumann eigenvalues and eigenfunctions for a rectangle Ω':

$$\lambda_n^N = \lambda_{\ell,m}^N = \pi^2 \left[\left(\frac{\ell}{a}\right)^2 + \left(\frac{m}{b}\right)^2 \right],$$
$$\varphi_n^N(x,y) = \varphi_{\ell,m}^N(x,y) = c_0 \cos\left(\frac{\pi \ell x}{a}\right) \cos\left(\frac{m\pi y}{b}\right). \quad n, \ell, m = 0, 1, 2, \ldots$$

where $c_0 := 2/\sqrt{ab}$.

- Clearly, the smallest eigenvalue is: $\lambda_0^N = \lambda_{0,0}^N = 0$, $\varphi_0^N(x,y) \equiv c_0$.
- How about the next smallest one? Since $a > b$,

$$\lambda_1^N = \lambda_{1,0}^N = \left(\frac{\pi}{a}\right)^2, \quad \varphi_1^N(x,y) = \varphi_{1,0}^N(x,y) = c_0 \cos\left(\frac{\pi}{a} x\right).$$

An Counterexample to the Domain Monotonicity ...

- Can easily compute the Neumann eigenvalues and eigenfunctions for a rectangle Ω':

$$\lambda_n^N = \lambda_{\ell,m}^N = \pi^2 \left[\left(\frac{\ell}{a}\right)^2 + \left(\frac{m}{b}\right)^2 \right],$$
$$\varphi_n^N(x,y) = \varphi_{\ell,m}^N(x,y) = c_0 \cos\left(\frac{\pi \ell x}{a}\right) \cos\left(\frac{m\pi y}{b}\right). \quad n, \ell, m = 0, 1, 2, \ldots$$

where $c_0 := 2/\sqrt{ab}$.

- Clearly, the smallest eigenvalue is: $\lambda_0^N = \lambda_{0,0}^N = 0$, $\varphi_0^N(x,y) \equiv c_0$.
- How about the next smallest one? Since $a > b$,

$$\lambda_1^N = \lambda_{1,0}^N = \left(\frac{\pi}{a}\right)^2, \quad \varphi_1^N(x,y) = \varphi_{1,0}^N(x,y) = c_0 \cos\left(\frac{\pi}{a} x\right).$$

An Counterexample to the Domain Monotonicity ...

- Can easily compute the Neumann eigenvalues and eigenfunctions for a rectangle Ω':

$$\lambda_n^N = \lambda_{\ell,m}^N = \pi^2 \left[\left(\frac{\ell}{a}\right)^2 + \left(\frac{m}{b}\right)^2 \right],$$
$$\varphi_n^N(x,y) = \varphi_{\ell,m}^N(x,y) = c_0 \cos\left(\frac{\pi \ell x}{a}\right) \cos\left(\frac{m\pi y}{b}\right). \quad n, \ell, m = 0, 1, 2, \ldots$$

  where $c_0 := 2/\sqrt{ab}$.

- Clearly, the smallest eigenvalue is: $\lambda_0^N = \lambda_{0,0}^N = 0$, $\varphi_0^N(x,y) \equiv c_0$.
- How about the next smallest one? Since $a > b$,

$$\lambda_1^N = \lambda_{1,0}^N = \left(\frac{\pi}{a}\right)^2, \quad \varphi_1^N(x,y) = \varphi_{1,0}^N(x,y) = c_0 \cos\left(\frac{\pi}{a} x\right).$$

An Counterexample to the Domain Monotonicity ...

- For λ_2^N, we have several possibilities, depending on the relationship between a and b.
- Here are just two examples:
  (i) If $\frac{2}{a} > \frac{1}{b}$, i.e., $b < a < 2b$, then

$$\lambda_2^N = \lambda_{0,1}^N = \left(\frac{\pi}{b}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{0,1}^N(x,y) = c_0 \cos\left(\frac{\pi}{b} y\right).$$

  (ii) If $\frac{2}{a} < \frac{1}{b}$, i.e., $a > 2b$, then

$$\lambda_2^N = \lambda_{2,0}^N = \left(\frac{2\pi}{a}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{2,0}^N(x,y) = c_0 \cos\left(\frac{2\pi}{a} x\right).$$

- The point is that λ_1^N of Ω' *only* depends on the *longer* side of the rectangle, in this case a.
- Now the *longer* side of Ω is equal to $\sqrt{(a-\alpha)^2 + (b-\beta)^2}$. By choosing appropriate $\alpha > 0$, $\beta > 0$ we can have $\sqrt{(a-\alpha)^2 + (b-\beta)^2} > a$. In other words, we can have $\lambda_1^N(\Omega) < \lambda_1^N(\Omega')$, even if $\Omega \subset \Omega'$.

An Counterexample to the Domain Monotonicity ...

- For λ_2^N, we have several possibilities, depending on the relationship between a and b.
- Here are just two examples:
  (i) If $\frac{2}{a} > \frac{1}{b}$, i.e., $b < a < 2b$, then
  $$\lambda_2^N = \lambda_{0,1}^N = \left(\frac{\pi}{b}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{0,1}^N(x,y) = c_0 \cos\left(\frac{\pi}{b} y\right).$$
  (ii) If $\frac{2}{a} < \frac{1}{b}$, i.e., $a > 2b$, then
  $$\lambda_2^N = \lambda_{2,0}^N = \left(\frac{2\pi}{a}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{2,0}^N(x,y) = c_0 \cos\left(\frac{2\pi}{a} x\right).$$
- The point is that λ_1^N of Ω' *only* depends on the *longer* side of the rectangle, in this case a.
- Now the *longer* side of Ω is equal to $\sqrt{(a-\alpha)^2 + (b-\beta)^2}$. By choosing appropriate $\alpha > 0$, $\beta > 0$ we can have $\sqrt{(a-\alpha)^2 + (b-\beta)^2} > a$. In other words, we can have $\lambda_1^N(\Omega) < \lambda_1^N(\Omega')$, even if $\Omega \subset \Omega'$.

An Counterexample to the Domain Monotonicity ...

- For λ_2^N, we have several possibilities, depending on the relationship between a and b.
- Here are just two examples:
  (i) If $\frac{2}{a} > \frac{1}{b}$, i.e., $b < a < 2b$, then

$$\lambda_2^N = \lambda_{0,1}^N = \left(\frac{\pi}{b}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{0,1}^N(x,y) = c_0 \cos\left(\frac{\pi}{b} y\right).$$

  (ii) If $\frac{2}{a} < \frac{1}{b}$, i.e., $a > 2b$, then

$$\lambda_2^N = \lambda_{2,0}^N = \left(\frac{2\pi}{a}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{2,0}^N(x,y) = c_0 \cos\left(\frac{2\pi}{a} x\right).$$

- The point is that λ_1^N of Ω' *only* depends on the *longer* side of the rectangle, in this case a.
- Now the *longer* side of Ω is equal to $\sqrt{(a-\alpha)^2 + (b-\beta)^2}$. By choosing appropriate $\alpha > 0$, $\beta > 0$ we can have $\sqrt{(a-\alpha)^2 + (b-\beta)^2} > a$. In other words, we can have $\lambda_1^N(\Omega) < \lambda_1^N(\Omega')$, even if $\Omega \subset \Omega'$.

An Counterexample to the Domain Monotonicity . . .

- For λ_2^N, we have several possibilities, depending on the relationship between a and b.
- Here are just two examples:
  (i) If $\frac{2}{a} > \frac{1}{b}$, i.e., $b < a < 2b$, then
  $$\lambda_2^N = \lambda_{0,1}^N = \left(\frac{\pi}{b}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{0,1}^N(x,y) = c_0 \cos\left(\frac{\pi}{b} y\right).$$
  (ii) If $\frac{2}{a} < \frac{1}{b}$, i.e., $a > 2b$, then
  $$\lambda_2^N = \lambda_{2,0}^N = \left(\frac{2\pi}{a}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{2,0}^N(x,y) = c_0 \cos\left(\frac{2\pi}{a} x\right).$$
- The point is that λ_1^N of Ω' *only* depends on the *longer* side of the rectangle, in this case a.
- Now the *longer* side of Ω is equal to $\sqrt{(a-\alpha)^2 + (b-\beta)^2}$. By choosing appropriate $\alpha > 0$, $\beta > 0$ we can have $\sqrt{(a-\alpha)^2 + (b-\beta)^2} > a$. In other words, we can have $\lambda_1^N(\Omega) < \lambda_1^N(\Omega')$, even if $\Omega \subset \Omega'$.

An Counterexample to the Domain Monotonicity ...

- For λ_2^N, we have several possibilities, depending on the relationship between a and b.
- Here are just two examples:
  (i) If $\frac{2}{a} > \frac{1}{b}$, i.e., $b < a < 2b$, then

$$\lambda_2^N = \lambda_{0,1}^N = \left(\frac{\pi}{b}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{0,1}^N(x,y) = c_0 \cos\left(\frac{\pi}{b} y\right).$$

  (ii) If $\frac{2}{a} < \frac{1}{b}$, i.e., $a > 2b$, then

$$\lambda_2^N = \lambda_{2,0}^N = \left(\frac{2\pi}{a}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{2,0}^N(x,y) = c_0 \cos\left(\frac{2\pi}{a} x\right).$$

- The point is that λ_1^N of Ω' *only* depends on the *longer* side of the rectangle, in this case a.
- Now the *longer* side of Ω is equal to $\sqrt{(a-\alpha)^2 + (b-\beta)^2}$. By choosing appropriate $\alpha > 0$, $\beta > 0$ we can have $\sqrt{(a-\alpha)^2 + (b-\beta)^2} > a$. In other words, we can have $\lambda_1^N(\Omega) < \lambda_1^N(\Omega')$, even if $\Omega \subset \Omega'$.

An Counterexample to the Domain Monotonicity ...

- For λ_2^N, we have several possibilities, depending on the relationship between a and b.
- Here are just two examples:
  (i) If $\frac{2}{a} > \frac{1}{b}$, i.e., $b < a < 2b$, then

$$\lambda_2^N = \lambda_{0,1}^N = \left(\frac{\pi}{b}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{0,1}^N(x,y) = c_0 \cos\left(\frac{\pi}{b} y\right).$$

  (ii) If $\frac{2}{a} < \frac{1}{b}$, i.e., $a > 2b$, then

$$\lambda_2^N = \lambda_{2,0}^N = \left(\frac{2\pi}{a}\right)^2, \quad \varphi_2^N(x,y) = \varphi_{2,0}^N(x,y) = c_0 \cos\left(\frac{2\pi}{a} x\right).$$

- The point is that λ_1^N of Ω' *only* depends on the *longer* side of the rectangle, in this case a.
- Now the *longer* side of Ω is equal to $\sqrt{(a-\alpha)^2 + (b-\beta)^2}$. By choosing appropriate $\alpha > 0$, $\beta > 0$ we can have $\sqrt{(a-\alpha)^2 + (b-\beta)^2} > a$. In other words, we can have $\lambda_1^N(\Omega) < \lambda_1^N(\Omega')$, even if $\Omega \subset \Omega'$.

Outline

Numerical Methods for Laplacian Eigenvalue Problems

- Finite Difference Method (FDM)
- Finite Element Method (FEM)
- Boundary Element Method (BEM)
- Radial Basis Functions (RBFs)
- *Method of Particular Solutions (MPS)*
  - Fox/Henrich/Moler 1967, Betcke/Trefethen 2005, Barnett 2009
- *Method of Fundamental Solutions (MFS)*
  - Trefftz 1926, ..., Karageorghis 2001, Alves/Antunes 2005, ...
- *Diagonalization of Integral Operators Commuting with Laplacian (NS, 2008)*
- ...

Numerical Methods for Laplacian Eigenvalue Problems

- Finite Difference Method (FDM)
- Finite Element Method (FEM)
- Boundary Element Method (BEM)
- Radial Basis Functions (RBFs)
- *Method of Particular Solutions (MPS)*
  - Fox/Henrich/Moler 1967, Betcke/Trefethen 2005, Barnett 2009
- *Method of Fundamental Solutions (MFS)*
  - Trefftz 1926, . . . , Karageorghis 2001, Alves/Antunes 2005, . . .
- *Diagonalization of Integral Operators Commuting with Laplacian (NS, 2008)*
- . . .

Numerical Methods for Laplacian Eigenvalue Problems

- Finite Difference Method (FDM)
- Finite Element Method (FEM)
- Boundary Element Method (BEM)
- Radial Basis Functions (RBFs)
- *Method of Particular Solutions (MPS)*
  - Fox/Henrich/Moler 1967, Betcke/Trefethen 2005, Barnett 2009
- *Method of Fundamental Solutions (MFS)*
  - Trefftz 1926, ..., Karageorghis 2001, Alves/Antunes 2005, ...
- *Diagonalization of Integral Operators Commuting with Laplacian (NS, 2008)*
- ...

Numerical Methods for Laplacian Eigenvalue Problems

- Finite Difference Method (FDM)
- Finite Element Method (FEM)
- Boundary Element Method (BEM)
- Radial Basis Functions (RBFs)
- *Method of Particular Solutions (MPS)*
  - Fox/Henrich/Moler 1967, Betcke/Trefethen 2005, Barnett 2009
- *Method of Fundamental Solutions (MFS)*
  - Trefftz 1926, ..., Karageorghis 2001, Alves/Antunes 2005, ...
- *Diagonalization of Integral Operators Commuting with Laplacian (NS, 2008)*
- ...

Numerical Methods for Laplacian Eigenvalue Problems

- Finite Difference Method (FDM)
- Finite Element Method (FEM)
- Boundary Element Method (BEM)
- Radial Basis Functions (RBFs)
- *Method of Particular Solutions (MPS)*
  - Fox/Henrich/Moler 1967, Betcke/Trefethen 2005, Barnett 2009
- *Method of Fundamental Solutions (MFS)*
  - Trefftz 1926, . . . , Karageorghis 2001, Alves/Antunes 2005, . . .
- *Diagonalization of Integral Operators Commuting with Laplacian (NS, 2008)*
- . . .

Numerical Methods for Laplacian Eigenvalue Problems

- Finite Difference Method (FDM)
- Finite Element Method (FEM)
- Boundary Element Method (BEM)
- Radial Basis Functions (RBFs)
- *Method of Particular Solutions (MPS)*
  - Fox/Henrich/Moler 1967, Betcke/Trefethen 2005, Barnett 2009
- *Method of Fundamental Solutions (MFS)*
  - Trefftz 1926, . . . , Karageorghis 2001, Alves/Antunes 2005, . . .
- *Diagonalization of Integral Operators Commuting with Laplacian (NS, 2008)*
- . . .

Numerical Methods for Laplacian Eigenvalue Problems

- Finite Difference Method (FDM)
- Finite Element Method (FEM)
- Boundary Element Method (BEM)
- Radial Basis Functions (RBFs)
- *Method of Particular Solutions (MPS)*
  - Fox/Henrich/Moler 1967, Betcke/Trefethen 2005, Barnett 2009
- *Method of Fundamental Solutions (MFS)*
  - Trefftz 1926, . . . , Karageorghis 2001, Alves/Antunes 2005, . . .
- *Diagonalization of Integral Operators Commuting with Laplacian (NS, 2008)*
- . . .

Numerical Methods for Laplacian Eigenvalue Problems

- Finite Difference Method (FDM)
- Finite Element Method (FEM)
- Boundary Element Method (BEM)
- Radial Basis Functions (RBFs)
- *Method of Particular Solutions (MPS)*
  - Fox/Henrich/Moler 1967, Betcke/Trefethen 2005, Barnett 2009
- *Method of Fundamental Solutions (MFS)*
  - Trefftz 1926, . . . , Karageorghis 2001, Alves/Antunes 2005, . . .
- *Diagonalization of Integral Operators Commuting with Laplacian (NS, 2008)*
- . . .

Numerical Methods for Laplacian Eigenvalue Problems

- Finite Difference Method (FDM)
- Finite Element Method (FEM)
- Boundary Element Method (BEM)
- Radial Basis Functions (RBFs)
- *Method of Particular Solutions (MPS)*
  - Fox/Henrich/Moler 1967, Betcke/Trefethen 2005, Barnett 2009
- *Method of Fundamental Solutions (MFS)*
  - Trefftz 1926, . . . , Karageorghis 2001, Alves/Antunes 2005, . . .
- *Diagonalization of Integral Operators Commuting with Laplacian (NS, 2008)*
- . . .

Numerical Methods for Laplacian Eigenvalue Problems

- Finite Difference Method (FDM)
- Finite Element Method (FEM)
- Boundary Element Method (BEM)
- Radial Basis Functions (RBFs)
- *Method of Particular Solutions (MPS)*
    - Fox/Henrich/Moler 1967, Betcke/Trefethen 2005, Barnett 2009
- *Method of Fundamental Solutions (MFS)*
    - Trefftz 1926, . . . , Karageorghis 2001, Alves/Antunes 2005, . . .
- *Diagonalization of Integral Operators Commuting with Laplacian (NS, 2008)*
- . . .

Outline

Recap on Difficulties Dealing with Laplacian

- Analysis of the Laplacian $\mathscr{L} = -\Delta$ is difficult due to its unboundedness, etc.
- Computing the eigenfunctions of $\mathscr{L}$ by directly solving the Helmholtz equation (or eigenvalue problem) on a general domain is tough.
- Much better to analyze its inverse, i.e., the Green's operator because it is compact and self-adjoint.
- Unfortunately, computing the Green's function for a general Ω satisfying the usual boundary condition (i.e., Dirichlet, Neumann) is also very difficult.

Recap on Difficulties Dealing with Laplacian

- Analysis of the Laplacian $\mathscr{L} = -\Delta$ is difficult due to its unboundedness, etc.
- Computing the eigenfunctions of $\mathscr{L}$ by directly solving the Helmholtz equation (or eigenvalue problem) on a general domain is tough.
- Much better to analyze its inverse, i.e., the Green's operator because it is compact and self-adjoint.
- Unfortunately, computing the Green's function for a general Ω satisfying the usual boundary condition (i.e., Dirichlet, Neumann) is also very difficult.

Recap on Difficulties Dealing with Laplacian

- Analysis of the Laplacian $\mathscr{L} = -\Delta$ is difficult due to its unboundedness, etc.
- Computing the eigenfunctions of $\mathscr{L}$ by directly solving the Helmholtz equation (or eigenvalue problem) on a general domain is tough.
- Much better to analyze its inverse, i.e., the Green's operator because it is compact and self-adjoint.
- Unfortunately, computing the Green's function for a general Ω satisfying the usual boundary condition (i.e., Dirichlet, Neumann) is also very difficult.

Recap on Difficulties Dealing with Laplacian

- Analysis of the Laplacian $\mathscr{L} = -\Delta$ is difficult due to its unboundedness, etc.
- Computing the eigenfunctions of $\mathscr{L}$ by directly solving the Helmholtz equation (or eigenvalue problem) on a general domain is tough.
- Much better to analyze its inverse, i.e., the Green's operator because it is compact and self-adjoint.
- Unfortunately, computing the Green's function for a general Ω satisfying the usual boundary condition (i.e., Dirichlet, Neumann) is also very difficult.

Integral Operators Commuting with Laplacian

- The key idea to avoid difficulties associated with the Laplacian $\mathscr{L}$ is to find an integral operator $\mathscr{K}$ commuting with $\mathscr{L}$ without imposing the strict boundary condition a priori.
- Then, we know that the eigenfunctions of $\mathscr{L}$ is the same as those of $\mathscr{K}$, which is easier to deal with, due to the following

Theorem (G. Frobenius 1896?; B. Friedman 1956)

Suppose $\mathscr{K}$ and $\mathscr{L}$ commute and one of them has an eigenvalue with finite multiplicity. Then, $\mathscr{K}$ and $\mathscr{L}$ share the same eigenfunction corresponding to that eigenvalue. That is, $\mathscr{L}\varphi = \lambda\varphi$ and $\mathscr{K}\varphi = \mu\varphi$.

Integral Operators Commuting with Laplacian

- The key idea to avoid difficulties associated with the Laplacian $\mathscr{L}$ is to find an integral operator $\mathscr{K}$ commuting with $\mathscr{L}$ without imposing the strict boundary condition a priori.
- Then, we know that the eigenfunctions of $\mathscr{L}$ is the same as those of $\mathscr{K}$, which is easier to deal with, due to the following

Theorem (G. Frobenius 1896?; B. Friedman 1956)

Suppose $\mathscr{K}$ and $\mathscr{L}$ commute and one of them has an eigenvalue with finite multiplicity. Then, $\mathscr{K}$ and $\mathscr{L}$ share the same eigenfunction corresponding to that eigenvalue. That is, $\mathscr{L}\varphi = \lambda\varphi$ and $\mathscr{K}\varphi = \mu\varphi$.

Integral Operators Commuting with Laplacian

- The key idea to avoid difficulties associated with the Laplacian $\mathscr{L}$ is to find an integral operator $\mathscr{K}$ commuting with $\mathscr{L}$ without imposing the strict boundary condition a priori.
- Then, we know that the eigenfunctions of $\mathscr{L}$ is the same as those of $\mathscr{K}$, which is easier to deal with, due to the following

Theorem (G. Frobenius 1896?; B. Friedman 1956)

Suppose $\mathscr{K}$ and $\mathscr{L}$ commute and one of them has an eigenvalue with finite multiplicity. Then, $\mathscr{K}$ and $\mathscr{L}$ share the same eigenfunction corresponding to that eigenvalue. That is, $\mathscr{L}\varphi = \lambda\varphi$ and $\mathscr{K}\varphi = \mu\varphi$.

Integral Operators Commuting with Laplacian . . .

- The inverse of $\mathscr{L}$ with some specific boundary condition (e.g., Dirichlet/Neumann/Robin) is also an integral operator whose kernel is called the *Green's function* $G(\boldsymbol{x}, \boldsymbol{y})$.
- Since it is not easy to obtain $G(\boldsymbol{x}, \boldsymbol{y})$ in general, let's replace $G(\boldsymbol{x}, \boldsymbol{y})$ by the fundamental solution of the Laplacian:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} -\frac{1}{2}|x - y| & \text{if } d = 1, \\ -\frac{1}{2\pi} \log|\boldsymbol{x} - \boldsymbol{y}| & \text{if } d = 2, \\ \frac{|\boldsymbol{x}-\boldsymbol{y}|^{2-d}}{(d-2)\omega_d} & \text{if } d > 2, \end{cases}$$

where $\omega_d := \frac{2\pi^{d/2}}{\Gamma(d/2)}$ is the surface area of the unit ball in $\mathbb{R}^d$, and $|\cdot|$ is the standard Euclidean norm.

- The price we pay is to have rather implicit, *non-local* boundary condition although we do not have to deal with this condition directly.

Integral Operators Commuting with Laplacian ...

- The inverse of $\mathscr{L}$ with some specific boundary condition (e.g., Dirichlet/Neumann/Robin) is also an integral operator whose kernel is called the *Green's function* $G(\boldsymbol{x}, \boldsymbol{y})$.
- Since it is not easy to obtain $G(\boldsymbol{x}, \boldsymbol{y})$ in general, let's replace $G(\boldsymbol{x}, \boldsymbol{y})$ by the fundamental solution of the Laplacian:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} -\frac{1}{2}|x - y| & \text{if } d = 1, \\ -\frac{1}{2\pi} \log|\boldsymbol{x} - \boldsymbol{y}| & \text{if } d = 2, \\ \frac{|\boldsymbol{x}-\boldsymbol{y}|^{2-d}}{(d-2)\omega_d} & \text{if } d > 2, \end{cases}$$

where $\omega_d := \frac{2\pi^{d/2}}{\Gamma(d/2)}$ is the surface area of the unit ball in $\mathbb{R}^d$, and $|\cdot|$ is the standard Euclidean norm.

- The price we pay is to have rather implicit, *non-local* boundary condition although we do not have to deal with this condition directly.

Integral Operators Commuting with Laplacian ...

- The inverse of $\mathscr{L}$ with some specific boundary condition (e.g., Dirichlet/Neumann/Robin) is also an integral operator whose kernel is called the *Green's function* $G(\boldsymbol{x}, \boldsymbol{y})$.
- Since it is not easy to obtain $G(\boldsymbol{x}, \boldsymbol{y})$ in general, let's replace $G(\boldsymbol{x}, \boldsymbol{y})$ by the fundamental solution of the Laplacian:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} -\frac{1}{2}|x-y| & \text{if } d=1, \\ -\frac{1}{2\pi}\log|\boldsymbol{x}-\boldsymbol{y}| & \text{if } d=2, \\ \frac{|\boldsymbol{x}-\boldsymbol{y}|^{2-d}}{(d-2)\omega_d} & \text{if } d>2, \end{cases}$$

  where $\omega_d := \frac{2\pi^{d/2}}{\Gamma(d/2)}$ is the surface area of the unit ball in $\mathbb{R}^d$, and $|\cdot|$ is the standard Euclidean norm.
- The price we pay is to have rather implicit, *non-local* boundary condition although we do not have to deal with this condition directly.

Integral Operators Commuting with Laplacian ...

- Let $\mathscr{K}$ be the integral operator with its kernel $K(\boldsymbol{x},\boldsymbol{y})$:

$$\mathscr{K} f(\boldsymbol{x}) := \int_{\Omega} K(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}, \quad f \in L^2(\Omega).$$

Theorem (NS 2005, 2008)

The integral operator $\mathscr{K}$ commutes with the Laplacian $\mathscr{L} = -\Delta$ with the following non-local boundary condition:

$$\int_{\partial\Omega} K(\boldsymbol{x},\boldsymbol{y}) \frac{\partial\varphi}{\partial\nu_{\boldsymbol{y}}}(\boldsymbol{y})\,\mathrm{d}s(\boldsymbol{y}) = -\frac{1}{2}\varphi(\boldsymbol{x}) + \operatorname{pv}\int_{\partial\Omega} \frac{\partial K(\boldsymbol{x},\boldsymbol{y})}{\partial\nu_{\boldsymbol{y}}}\varphi(\boldsymbol{y})\,\mathrm{d}s(\boldsymbol{y}), \quad \forall \boldsymbol{x} \in \partial\Omega,$$

where φ is an eigenfunction common for both operators, and pv indicates the Cauchy principal value.

Integral Operators Commuting with Laplacian . . .

- Let $\mathscr{K}$ be the integral operator with its kernel $K(\boldsymbol{x}, \boldsymbol{y})$:

$$\mathscr{K} f(\boldsymbol{x}) := \int_{\Omega} K(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y}, \quad f \in L^2(\Omega).$$

Theorem (NS 2005, 2008)

The integral operator $\mathscr{K}$ commutes with the Laplacian $\mathscr{L} = -\Delta$ with the following non-local boundary condition:

$$\int_{\partial\Omega} K(\boldsymbol{x}, \boldsymbol{y}) \frac{\partial \varphi}{\partial \nu_{\boldsymbol{y}}}(\boldsymbol{y}) \, \mathrm{d}s(\boldsymbol{y}) = -\frac{1}{2}\varphi(\boldsymbol{x}) + \mathrm{pv}\int_{\partial\Omega} \frac{\partial K(\boldsymbol{x}, \boldsymbol{y})}{\partial \nu_{\boldsymbol{y}}} \varphi(\boldsymbol{y}) \, \mathrm{d}s(\boldsymbol{y}), \quad \forall \boldsymbol{x} \in \partial\Omega,$$

where φ is an eigenfunction common for both operators, and pv indicates the Cauchy principal value.

Integral Operators Commuting with Laplacian . . .

Corollary (NS 2009)

The eigenfunction $\varphi(\boldsymbol{x})$ of the integral operator $\mathcal{K}$ in the previous theorem can be extended outside the domain Ω and satisfies the following equation:

$$-\Delta\varphi = \begin{cases} \lambda\varphi & \text{if } \boldsymbol{x} \in \Omega; \\ 0 & \text{if } \boldsymbol{x} \in \mathbb{R}^d \setminus \overline{\Omega}, \end{cases}$$

with the boundary condition that φ and $\dfrac{\partial\varphi}{\partial\nu}$ are continuous across the boundary $\partial\Omega$. Moreover, as $|\boldsymbol{x}| \to \infty$, $\varphi(\boldsymbol{x})$ must be of the following form:

$$\varphi(\boldsymbol{x}) = \begin{cases} \text{const} \cdot |\boldsymbol{x}|^{2-d} + O\left(|\boldsymbol{x}|^{1-d}\right) & \text{if } d \neq 2; \\ \text{const} \cdot \ln|\boldsymbol{x}| + O\left(|\boldsymbol{x}|^{-1}\right) & \text{if } d = 2. \end{cases}$$

Integral Operators Commuting with Laplacian . . .

Corollary (NS 2005, 2008)

The integral operator $\mathscr{K}$ is compact and self-adjoint on $L^2(\Omega)$. Thus, the kernel $K(\boldsymbol{x},\boldsymbol{y})$ has the following eigenfunction expansion (in the sense of mean convergence):

$$K(\boldsymbol{x},\boldsymbol{y}) \sim \sum_{j=1}^{\infty} \mu_j \varphi_j(\boldsymbol{x})\overline{\varphi_j(\boldsymbol{y})},$$

and $\{\varphi_j\}_j$ forms an orthonormal basis of $L^2(\Omega)$.

Outline

1D Example

- Consider the unit interval $\Omega = (0,1)$.
- Then, our integral operator $\mathscr{K}$ with the kernel $K(x,y) = -|x-y|/2$ gives rise to the following eigenvalue problem:

$$-\varphi'' = \lambda\varphi, \quad x \in (0,1);$$

$$\varphi(0) + \varphi(1) = -\varphi'(0) = \varphi'(1).$$

- The kernel $K(x,y)$ is of *Toeplitz* form $\Longrightarrow$ Eigenvectors must have even and odd symmetry (Cantoni-Butler '76).
- In this case, we have the following explicit solution.

1D Example

- Consider the unit interval $\Omega = (0,1)$.
- Then, our integral operator $\mathscr{K}$ with the kernel $K(x,y) = -|x-y|/2$ gives rise to the following eigenvalue problem:

$$-\varphi'' = \lambda\varphi, \quad x \in (0,1);$$

$$\varphi(0) + \varphi(1) = -\varphi'(0) = \varphi'(1).$$

- The kernel $K(x,y)$ is of *Toeplitz* form $\Longrightarrow$ Eigenvectors must have even and odd symmetry (Cantoni-Butler '76).
- In this case, we have the following explicit solution.

1D Example

- Consider the unit interval $\Omega = (0,1)$.
- Then, our integral operator $\mathcal{K}$ with the kernel $K(x,y) = -|x-y|/2$ gives rise to the following eigenvalue problem:

$$-\varphi'' = \lambda\varphi, \quad x \in (0,1);$$

$$\varphi(0) + \varphi(1) = -\varphi'(0) = \varphi'(1).$$

- The kernel $K(\boldsymbol{x},\boldsymbol{y})$ is of *Toeplitz* form $\Longrightarrow$ Eigenvectors must have even and odd symmetry (Cantoni-Butler '76).
- In this case, we have the following explicit solution.

1D Example

- Consider the unit interval $\Omega = (0, 1)$.
- Then, our integral operator $\mathscr{K}$ with the kernel $K(x, y) = -|x - y|/2$ gives rise to the following eigenvalue problem:

$$-\varphi'' = \lambda\varphi, \quad x \in (0, 1);$$

$$\varphi(0) + \varphi(1) = -\varphi'(0) = \varphi'(1).$$

- The kernel $K(\boldsymbol{x}, \boldsymbol{y})$ is of *Toeplitz* form $\Longrightarrow$ Eigenvectors must have even and odd symmetry (Cantoni-Butler '76).
- In this case, we have the following explicit solution.

1D Example . . .

- $\lambda_0 \approx -5.756915$, which is a solution of $\tanh \frac{\sqrt{-\lambda_0}}{2} = \frac{2}{\sqrt{-\lambda_0}}$,

$$\varphi_0(x) = A_0 \cosh \sqrt{-\lambda_0}\left(x - \frac{1}{2}\right);$$

- $\lambda_{2m-1} = (2m-1)^2\pi^2, \; m = 1, 2, \ldots,$

$$\varphi_{2m-1}(x) = \sqrt{2}\cos(2m-1)\pi x;$$

- $\lambda_{2m}, \; m = 1, 2, \ldots,$ which are solutions of $\tan \frac{\sqrt{\lambda_{2m}}}{2} = -\frac{2}{\sqrt{\lambda_{2m}}}$,

$$\varphi_{2m}(x) = A_{2m} \cos \sqrt{\lambda_{2m}}\left(x - \frac{1}{2}\right),$$

where $A_k, \; k = 0, 1, \ldots$ are normalization constants.

1D Example ...

- $\lambda_0 \approx -5.756915$, which is a solution of $\tanh \frac{\sqrt{-\lambda_0}}{2} = \frac{2}{\sqrt{-\lambda_0}}$,

$$\varphi_0(x) = A_0 \cosh \sqrt{-\lambda_0}\left(x - \frac{1}{2}\right);$$

- $\lambda_{2m-1} = (2m-1)^2\pi^2,\ m = 1, 2, \ldots,$

$$\varphi_{2m-1}(x) = \sqrt{2}\cos(2m-1)\pi x;$$

- $\lambda_{2m},\ m = 1, 2, \ldots,$ which are solutions of $\tan \frac{\sqrt{\lambda_{2m}}}{2} = -\frac{2}{\sqrt{\lambda_{2m}}}$,

$$\varphi_{2m}(x) = A_{2m} \cos \sqrt{\lambda_{2m}}\left(x - \frac{1}{2}\right),$$

where $A_k,\ k = 0, 1, \ldots$ are normalization constants.

1D Example . . .

- $\lambda_0 \approx -5.756915$, which is a solution of $\tanh \frac{\sqrt{-\lambda_0}}{2} = \frac{2}{\sqrt{-\lambda_0}}$,

$$\varphi_0(x) = A_0 \cosh \sqrt{-\lambda_0}\left(x - \frac{1}{2}\right);$$

- $\lambda_{2m-1} = (2m-1)^2\pi^2,\ m = 1,2,\ldots,$

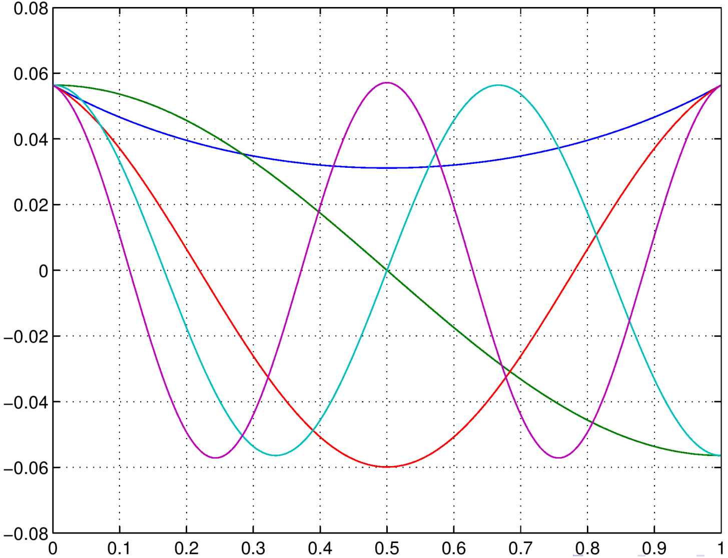
$$\varphi_{2m-1}(x) = \sqrt{2}\cos(2m-1)\pi x;$$

- $\lambda_{2m},\ m = 1,2,\ldots,$ which are solutions of $\tan \frac{\sqrt{\lambda_{2m}}}{2} = -\frac{2}{\sqrt{\lambda_{2m}}}$,

$$\varphi_{2m}(x) = A_{2m} \cos \sqrt{\lambda_{2m}}\left(x - \frac{1}{2}\right),$$

where $A_k,\ k = 0,1,\ldots$ are normalization constants.

First 5 Basis Functions



1D Example: Comparison

- The Laplacian eigenfunctions with the Dirichlet boundary condition: $-\varphi'' = \lambda\varphi$, $\varphi(0) = \varphi(1) = 0$, are *sines*. The Green's function in this case is:

$$G_D(x, y) = \min(x, y) - xy.$$

- Those with the Neumann boundary condition, i.e., $\varphi'(0) = \varphi'(1) = 0$, are *cosines*. The Green's function is:

$$G_N(x, y) = -\max(x, y) + \frac{1}{2}(x^2 + y^2) + \frac{1}{3}.$$

- Remark: Gridpoint ⇔ DST-I/DCT-I;
Midpoint⇔ DST-II/DCT-II.

1D Example: Comparison

- The Laplacian eigenfunctions with the Dirichlet boundary condition: $-\varphi'' = \lambda\varphi$, $\varphi(0) = \varphi(1) = 0$, are *sines*. The Green's function in this case is:
$$G_D(x, y) = \min(x, y) - xy.$$
- Those with the Neumann boundary condition, i.e., $\varphi'(0) = \varphi'(1) = 0$, are *cosines*. The Green's function is:
$$G_N(x, y) = -\max(x, y) + \frac{1}{2}(x^2 + y^2) + \frac{1}{3}.$$
- Remark: Gridpoint ⇔ DST-I/DCT-I; Midpoint⇔ DST-II/DCT-II.

1D Example: Comparison

- The Laplacian eigenfunctions with the Dirichlet boundary condition: $-\varphi'' = \lambda\varphi$, $\varphi(0) = \varphi(1) = 0$, are *sines*. The Green's function in this case is:
$$G_D(x, y) = \min(x, y) - xy.$$
- Those with the Neumann boundary condition, i.e., $\varphi'(0) = \varphi'(1) = 0$, are *cosines*. The Green's function is:
$$G_N(x, y) = -\max(x, y) + \frac{1}{2}(x^2 + y^2) + \frac{1}{3}.$$
- Remark: Gridpoint ⇔ DST-I/DCT-I;
Midpoint⇔ DST-II/DCT-II.

2D Example

- Consider the unit disk Ω. Then, our integral operator $\mathscr{K}$ with the kernel $K(\boldsymbol{x},\boldsymbol{y}) = -\frac{1}{2\pi}\log|\boldsymbol{x}-\boldsymbol{y}|$ gives rise to:

$$-\Delta\varphi = \lambda\varphi, \quad \text{in } \Omega;$$

$$\left.\frac{\partial\varphi}{\partial\nu}\right|_{\partial\Omega} = \left.\frac{\partial\varphi}{\partial r}\right|_{\partial\Omega} = -\left.\frac{\partial\mathscr{H}\varphi}{\partial\theta}\right|_{\partial\Omega},$$

where $\mathscr{H}$ is the Hilbert transform for the circle, i.e.,

$$\mathscr{H}f(\theta) := \frac{1}{2\pi}\,\mathrm{pv}\!\int_{-\pi}^{\pi} f(\eta)\cot\left(\frac{\theta-\eta}{2}\right)\mathrm{d}\eta \quad \theta\in[-\pi,\pi].$$

- Let $j_{k,\ell}$ is the ℓth zero of the Bessel function of order k, $J_k(j_{k,\ell})=0$. Then,

$$\varphi_{m,n}(r,\theta) = \begin{cases} J_m(j_{m-1,n}\,r)\binom{\cos}{\sin}(m\theta) & \text{if } m=1,2,\ldots,\ n=1,2,\ldots, \\ J_0(j_{0,n}\,r) & \text{if } m=0,\ n=1,2,\ldots, \end{cases}$$

$$\lambda_{m,n} = \begin{cases} j^2_{m-1,n}, & \text{if } m=1,\ldots,\ n=1,2,\ldots, \\ j^2_{0,n} & \text{if } m=0,\ n=1,2,\ldots. \end{cases}$$

2D Example

- Consider the unit disk Ω. Then, our integral operator $\mathscr{K}$ with the kernel $K(\boldsymbol{x},\boldsymbol{y}) = -\frac{1}{2\pi}\log|\boldsymbol{x}-\boldsymbol{y}|$ gives rise to:

$$-\Delta\varphi = \lambda\varphi, \quad \text{in } \Omega;$$

$$\left.\frac{\partial\varphi}{\partial\nu}\right|_{\partial\Omega} = \left.\frac{\partial\varphi}{\partial r}\right|_{\partial\Omega} = -\left.\frac{\partial\mathscr{H}\varphi}{\partial\theta}\right|_{\partial\Omega},$$

where $\mathscr{H}$ is the Hilbert transform for the circle, i.e.,

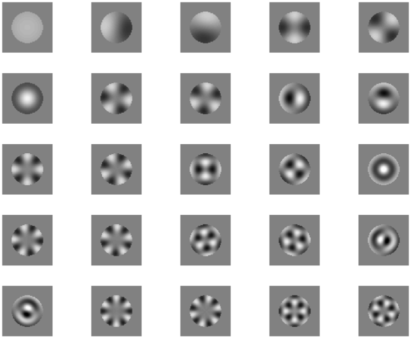
$$\mathscr{H}f(\theta) := \frac{1}{2\pi}\,\mathrm{pv}\int_{-\pi}^{\pi} f(\eta)\cot\left(\frac{\theta-\eta}{2}\right)\mathrm{d}\eta \quad \theta\in[-\pi,\pi].$$

- Let $j_{k,\ell}$ is the ℓth zero of the Bessel function of order k, $J_k(j_{k,\ell}) = 0$. Then,

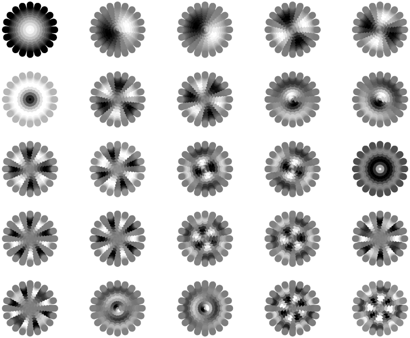
$$\varphi_{m,n}(r,\theta) = \begin{cases} J_m(j_{m-1,n}\,r)\binom{\cos}{\sin}(m\theta) & \text{if } m=1,2,\ldots,\ n=1,2,\ldots, \\ J_0(j_{0,n}\,r) & \text{if } m=0,\ n=1,2,\ldots, \end{cases}$$

$$\lambda_{m,n} = \begin{cases} j_{m-1,n}^2, & \text{if } m=1,\ldots,\ n=1,2,\ldots, \\ j_{0,n}^2 & \text{if } m=0,\ n=1,2,\ldots. \end{cases}$$

First 25 Basis Functions



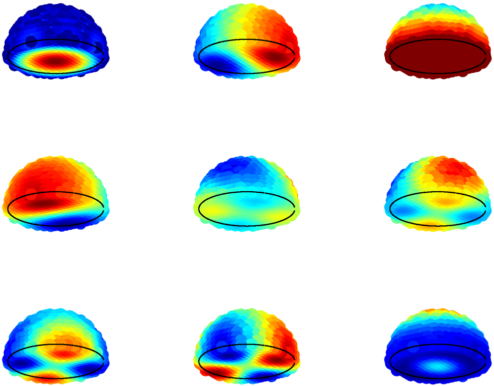
(a) Our Basis



(b) Dirichlet-Laplace

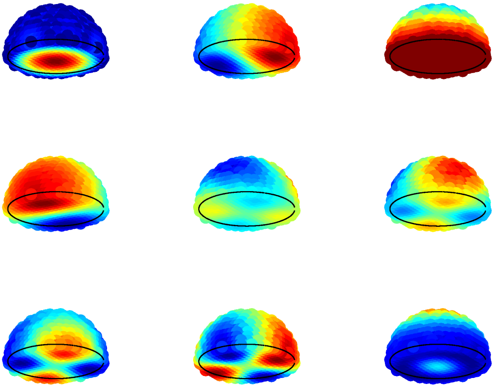
3D Example

- Consider the unit ball Ω in $\mathbb{R}^3$. Then, our integral operator $\mathscr{K}$ with the kernel $K(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{4\pi|\boldsymbol{x}-\boldsymbol{y}|}$.
- Top 9 eigenfunctions cut at the equator viewed from the south:



3D Example

- Consider the unit ball Ω in $\mathbb{R}^3$. Then, our integral operator $\mathscr{K}$ with the kernel $K(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{4\pi|\boldsymbol{x}-\boldsymbol{y}|}$.
- Top 9 eigenfunctions cut at the equator viewed from the south:



Outline

Discretization of the Problem

- Assume that the whole dataset consists of a collection of data sampled on a regular grid, and that each sampling cell is a box of size $\prod_{i=1}^{d} \Delta x_i$.
- Assume that an object of our interest Ω consists of a subset of these boxes whose centers are $\{\boldsymbol{x}_i\}_{i=1}^{N}$.
- Under these assumptions, we can approximate the integral eigenvalue problem $\mathscr{K}\varphi = \mu\varphi$ with a simple quadrature rule with node-weight pairs $(\boldsymbol{x}_j, w_j)$ as follows.

$$\sum_{j=1}^{N} w_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)\varphi(\boldsymbol{x}_j) = \mu\varphi(\boldsymbol{x}_i), \quad i = 1, \ldots, N, \quad w_j = \prod_{i=1}^{d} \Delta x_i.$$

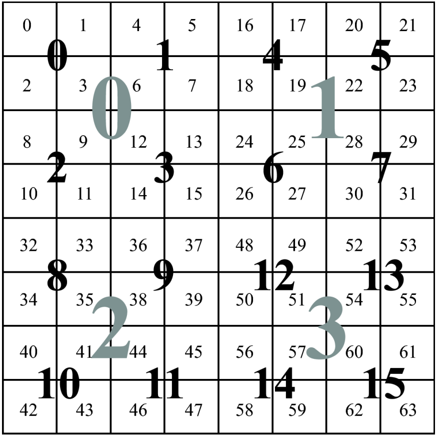
- Let $K_{i,j} := w_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\varphi_i := \varphi(\boldsymbol{x}_i)$, and $\boldsymbol{\varphi} := (\varphi_1, \ldots, \varphi_N)^{\mathsf{T}} \in \mathbb{R}^N$. Then, the above equation can be written in a matrix-vector format as: $K\boldsymbol{\varphi} = \mu\boldsymbol{\varphi}$, where $K = (K_{ij}) \in \mathbb{R}^{N \times N}$. Under our assumptions, the weight w_j does not depend on j, which makes K symmetric.

Outline

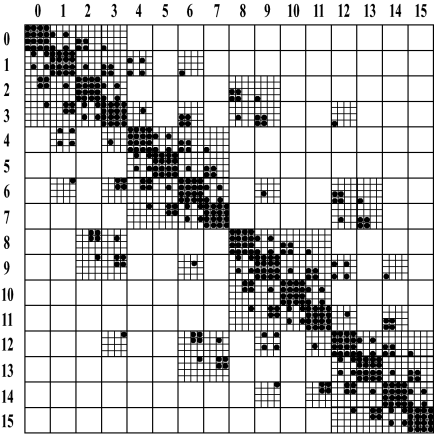
A Possible Fast Algorithm for Computing φ_j's

- Observation: our kernel function $K(\boldsymbol{x}, \boldsymbol{y})$ is of special form, i.e., the fundamental solution of Laplacian used in potential theory.
- Idea: Accelerate the matrix-vector product $K\boldsymbol{\varphi}$ using the Fast Multipole Method (FMM).
- Convert the kernel matrix to the tree-structured matrix via the FMM whose submatrices are nicely organized in terms of their ranks. (Computational cost: our current implementation costs $O(N^2)$, but can achieve $O(N \log N)$ via the randomized SVD algorithm of Woolfe-Liberty-Rokhlin-Tygert (2008)).
- Construct $O(N)$ matrix-vector product module fully utilizing rank information (See also the work of Bremer (2007) and the "HSS" algorithm of Chandrasekaran et al. (2006)).
- Embed that matrix-vector product module in the Krylov subspace method, e.g., Lanczos iteration.
  (Computational cost: $O(N)$ for each eigenvalue/eigenvector).

Tree-Structured Matrix via FMM

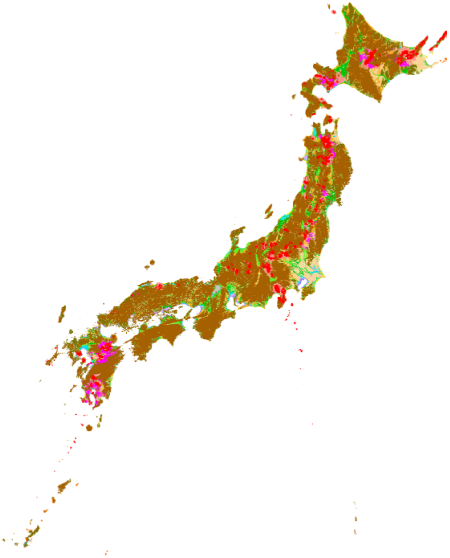


(a) Hierarchical indexing scheme

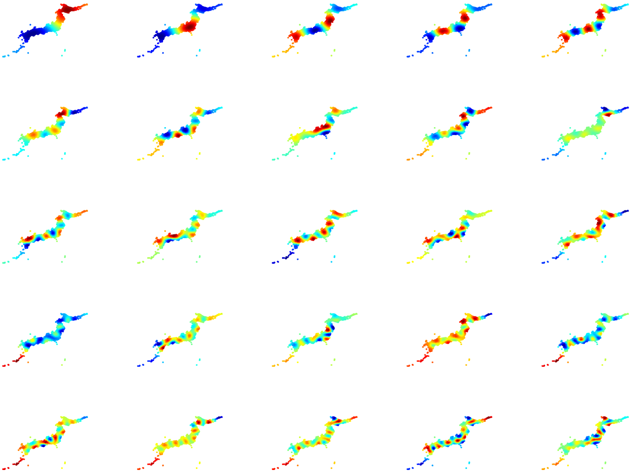


(b) Tree-Structured Matrix

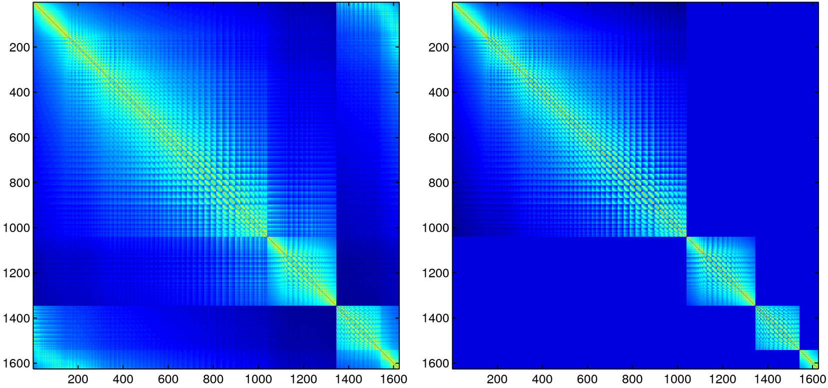
A Real Challenge: Kernel matrix is of 387924×387924.



First 25 Basis Functions via the FMM-based algorithm



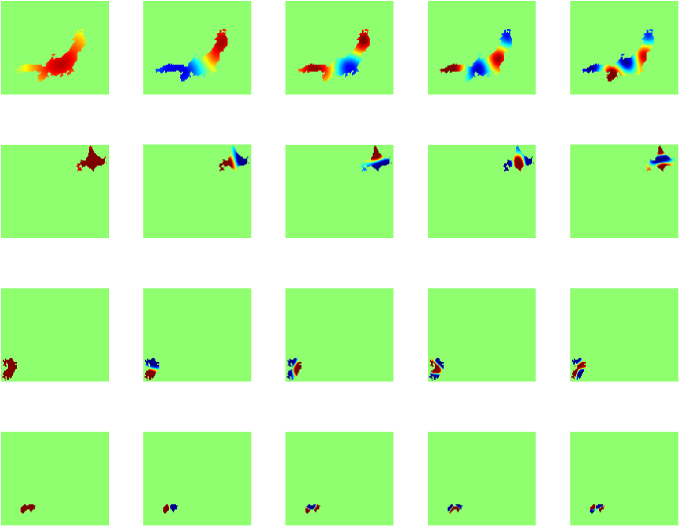
Splitting into Subproblems for Faster Computation



(a) Whole islands

(b) Separated islands

Eigenfunctions for Separated Islands



Outline

General Comments on Applications

Laplacian eigenfunctions on an irregular domain should be useful for:

- Interactive image analysis, discrimination, interpretation:
    - Medical image analysis: e.g., hippocampal shape analysis for early Alzheimer's
    - Biometry: e.g., identification and characterization of eyes, faces, etc.
- Geophysical data assimilation:
    - Incorporating ocean current data measured by high frequency radar into a numerical model;
    - Interpolation, extrapolation, prediction of vector-valued meteorology data (temperature, pressure, wind speed, etc.) measured at the weather station in the 3D terrain.
- ...

Remark on the DC vector

- The Laplacian eigenfunction with the least oscillation computed by diagonalizing the commuting integral operator is *not* the constant (i.e., *DC*) vector $\boldsymbol{\chi}_\Omega := \mathbf{1}_N / \sqrt{N} \in \mathbb{R}^N$.
- If some application needs to have the DC vector of a given domain Ω and the basis vectors orthogonal to the DC vector, there is a way to include the DC vector into the picture.
- Consider the *orthogonal complement* to the 1D subspace $\mathrm{span}\{\chi_\Omega\}$ in the column space of the kernel matrix K:

$$\widetilde{K} = \left(I - \chi_\Omega \chi_\Omega^\top\right) K.$$

- Then, χ_Ω together with the eigenvectors of $\widetilde{K}$ corresponding to the largest $N-1$ eigenvalues form the desired orthonormal basis.

Remark on the DC vector

- The Laplacian eigenfunction with the least oscillation computed by diagonalizing the commuting integral operator is *not* the constant (i.e., *DC*) vector $\boldsymbol{\chi}_\Omega := \mathbf{1}_N / \sqrt{N} \in \mathbb{R}^N$.
- If some application needs to have the DC vector of a given domain Ω and the basis vectors orthogonal to the DC vector, there is a way to include the DC vector into the picture.
- Consider the *orthogonal complement* to the 1D subspace span$\{\chi_\Omega\}$ in the column space of the kernel matrix K:

$$\widetilde{K} = \left(I - \chi_\Omega \chi_\Omega^{\mathsf{T}}\right) K.$$

- Then, χ_Ω together with the eigenvectors of $\widetilde{K}$ corresponding to the largest $N-1$ eigenvalues form the desired orthonormal basis.

Remark on the DC vector

- The Laplacian eigenfunction with the least oscillation computed by diagonalizing the commuting integral operator is *not* the constant (i.e., *DC*) vector $\boldsymbol{\chi}_\Omega := \mathbf{1}_N / \sqrt{N} \in \mathbb{R}^N$.
- If some application needs to have the DC vector of a given domain Ω and the basis vectors orthogonal to the DC vector, there is a way to include the DC vector into the picture.
- Consider the *orthogonal complement* to the 1D subspace $\mathrm{span}\{\boldsymbol{\chi}_\Omega\}$ in the column space of the kernel matrix K:

$$\widetilde{K} = \left(I - \boldsymbol{\chi}_\Omega \boldsymbol{\chi}_\Omega^{\mathsf{T}}\right) K.$$

- Then, χ_Ω together with the eigenvectors of $\widetilde{K}$ corresponding to the largest $N-1$ eigenvalues form the desired orthonormal basis.

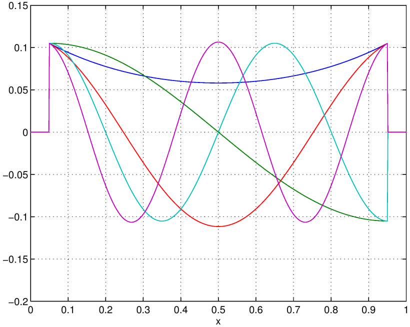
Remark on the DC vector

- The Laplacian eigenfunction with the least oscillation computed by diagonalizing the commuting integral operator is *not* the constant (i.e., *DC*) vector $\boldsymbol{\chi}_\Omega := \mathbf{1}_N / \sqrt{N} \in \mathbb{R}^N$.
- If some application needs to have the DC vector of a given domain Ω and the basis vectors orthogonal to the DC vector, there is a way to include the DC vector into the picture.
- Consider the *orthogonal complement* to the 1D subspace $\mathrm{span}\{\boldsymbol{\chi}_\Omega\}$ in the column space of the kernel matrix K:

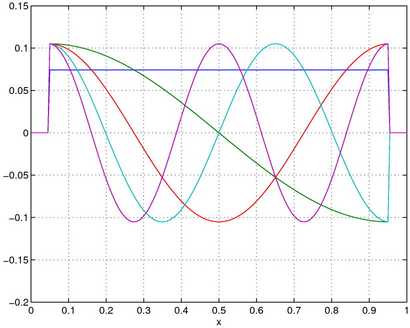
$$\widetilde{K} = \left(I - \boldsymbol{\chi}_\Omega \boldsymbol{\chi}_\Omega^{\mathsf{T}}\right) K.$$

- Then, $\boldsymbol{\chi}_\Omega$ together with the eigenvectors of $\widetilde{K}$ corresponding to the largest $N-1$ eigenvalues form the desired orthonormal basis.

Remark on the DC vector . . .



(a) Laplacian Eigenfunctions via Commuting Integral Operator

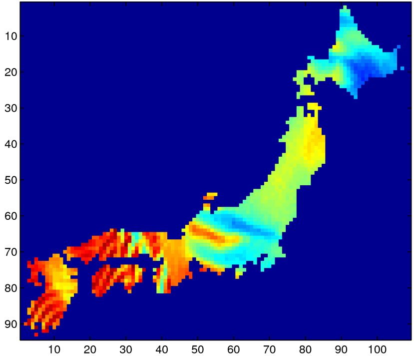


(b) Laplacian Eigenfunctions incorporating the DC vector

$\Longrightarrow$ leads to the *generalized discrete cosine basis!*

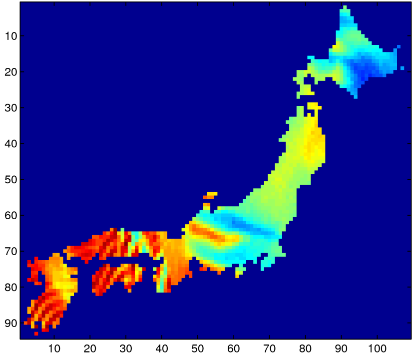
Outline

Image Approximation; Comparison with Wavelets

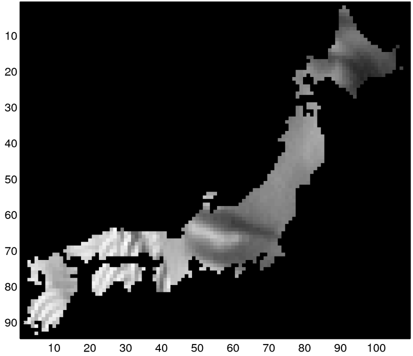


(a) What data?

Image Approximation; Comparison with Wavelets

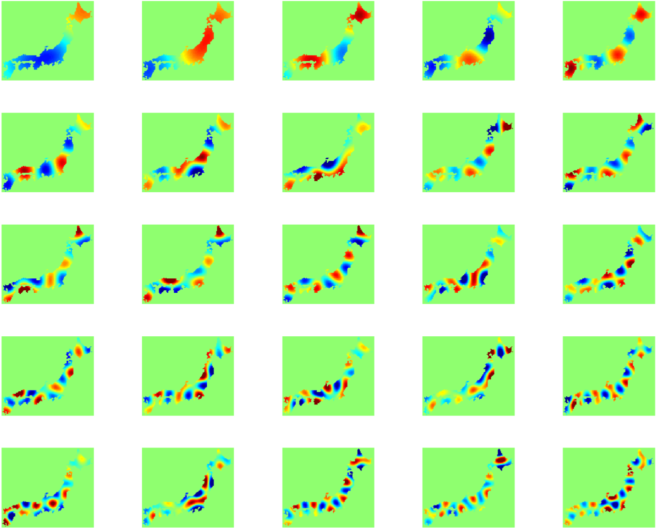


(a) What data?

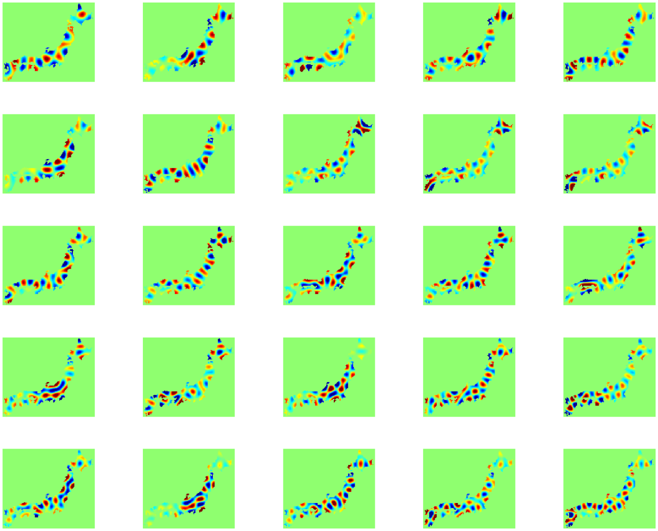


(b) $\chi_J \cdot$ Barbara

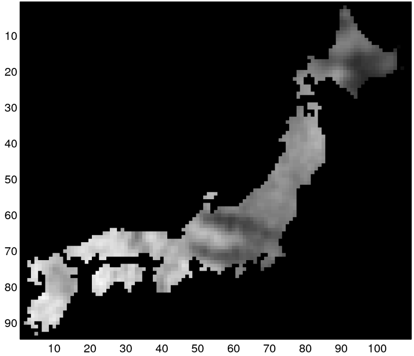
First 25 Basis Functions



Next 25 Basis Functions

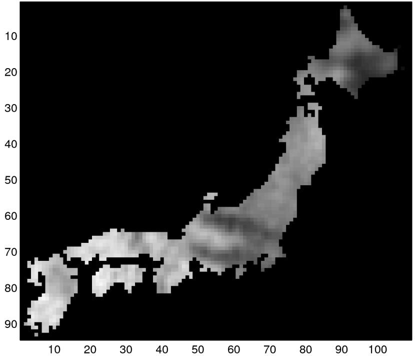


Reconstruction with Top 100 Coefficients

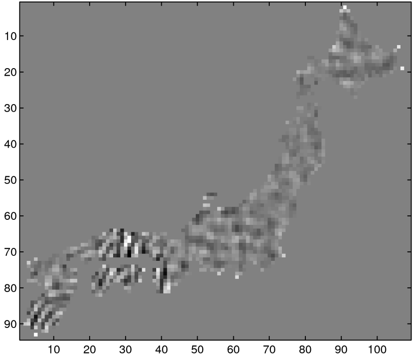


(a) Reconstruction

Reconstruction with Top 100 Coefficients

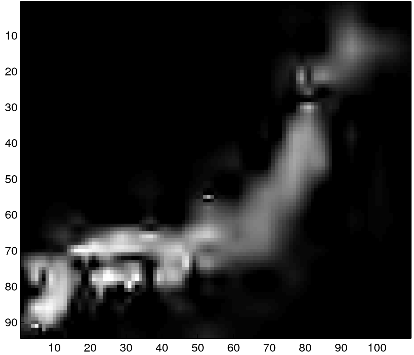


(a) Reconstruction



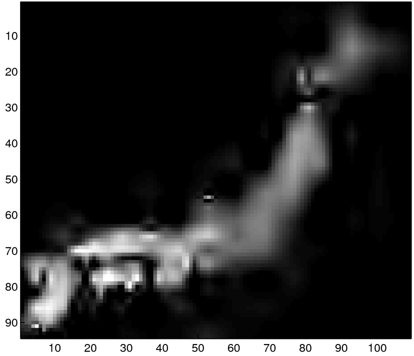
(b) Error

Reconstruction with Top 100 2D Wavelets (Symmlet 8)

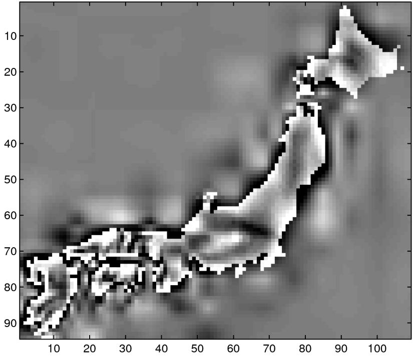


(a) Reconstruction

Reconstruction with Top 100 2D Wavelets (Symmlet 8)

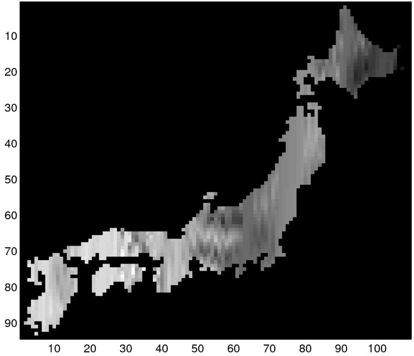


(a) Reconstruction



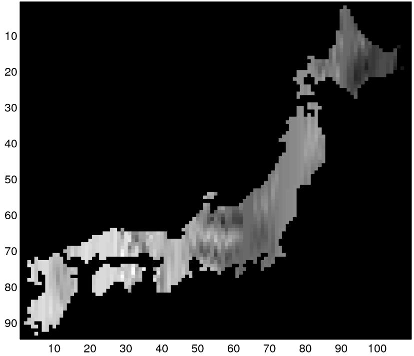
(b) Error

Reconstruction with Top 100 1D Wavelets (Symmlet 8)

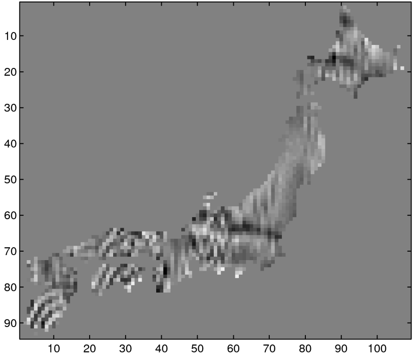


(a) Reconstruction

Reconstruction with Top 100 1D Wavelets (Symmlet 8)

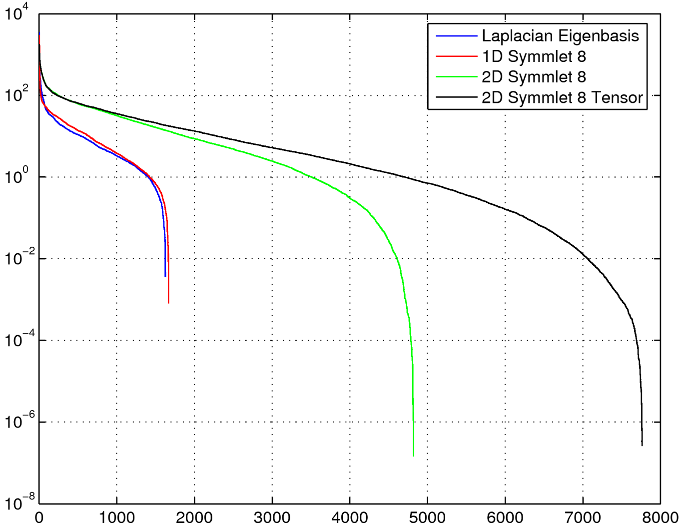


(a) Reconstruction



(b) Error

Comparison of Coefficient Decay



Outline

Experiments on Domains with Perturbed Boundaries

We will use the following domains for our experiments:

Ω_1: The Japanese Islands

Ω_2: A smoothed and connected version of Ω_1;

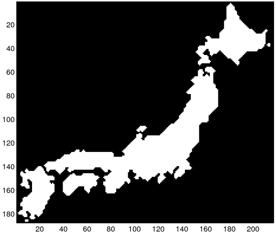
Ω_3: The same as Ω_2 but with a "jaggy" boundary curve

Ω_4: The two-component version of Ω_2.

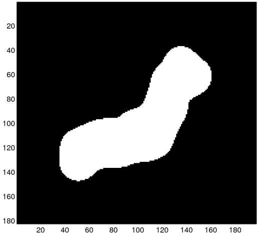
As for the data on these domains, we adopted three functions with different smoothness:

1. A discontinuous function (i.e., a simple step function whose discontinuity is a straight line along the "spine" or the main axis of the domain);
2. A pyramid-shaped function, which is continuous and its first order partial derivatives are of bounded variation;
3. The standard Gaussian function.

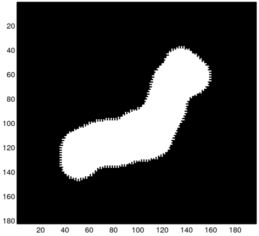
The Domains with Perturbed Boundaries



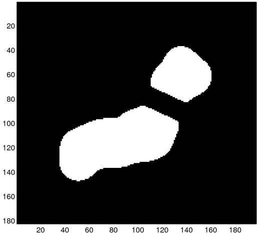
(a) χ_{Ω_1}



(b) χ_{Ω_2}

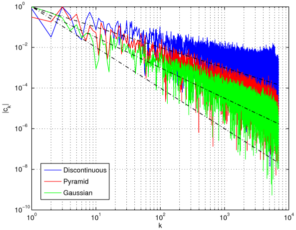


(c) χ_{Ω_3}

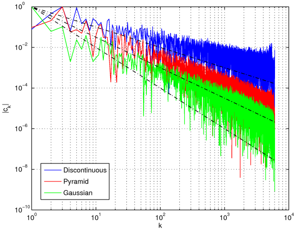


(d) χ_{Ω_4}

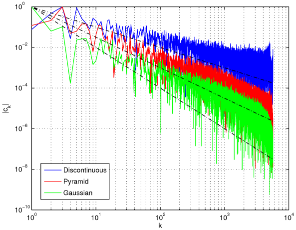
Decay Rates of the Expansion Coefficients (Unsorted)



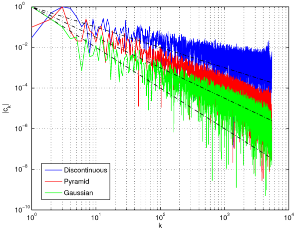
(a) Decay rates on Ω_1



(b) Decay rates on Ω_2



(c) Decay rates on Ω_3

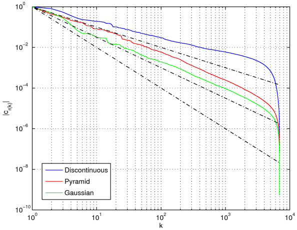


(d) Decay rates on Ω_4

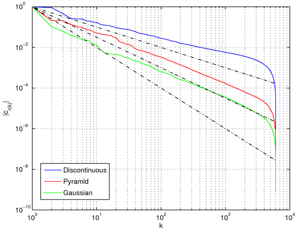
Observations on the Decay Rates

- The decay rates reflect the intrinsic smoothness of the functions living in the domain, but are not affected by the existence of the boundary of the domains.
- The decay rates are rather insensitive to the smoothness of the boundary curves. In particular, the plots for Ω_2, Ω_3, and Ω_4 are virtually the same whereas those for Ω_1—the most complicated domain among these four—seem slightly worse than the others. Yet all behave better than $O(k^{-1})$.
- The decay rates are rather insensitive to the number of the separated subdomains. Again, it will be also of interest to investigate the behavior the conventional Laplacian eigenfunctions in this respect.
- Although the coefficient plots oscillate around the linear lines (in the log-log scale), the decay rates $O(k^{-\alpha})$, regardless of the domain shapes, behave as follows. For the discontinuous functions, $\alpha < 1$. For the pyramid-shape function, $1 < \alpha < 1.5$. For the Gaussian function, $\alpha \geq 1.5$.

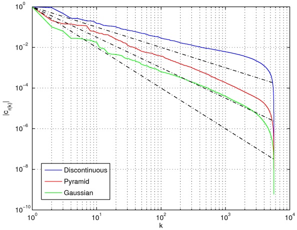
Decay Rates of the Expansion Coefficients (Sorted)



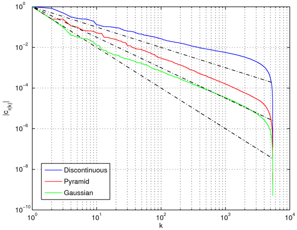
(a) Decay rates on Ω_1



(b) Decay rates on Ω_2



(c) Decay rates on Ω_3



(d) Decay rates on Ω_4

Conjecture on the Coefficient Decay Rate

Conjecture (NS 2007)

Let Ω be a C^2-domain of general shape and let $f \in C\left(\overline{\Omega}\right)$ with $\dfrac{\partial f}{\partial x_j} \in BV\left(\overline{\Omega}\right)$ for $j = 1, \ldots, d$. Let $\{c_k = \langle f, \varphi_k \rangle\}_{k \in \mathbb{N}}$ be the expansion coefficients of f with respect to our Laplacian eigenbasis on this domain. Then, $|c_k|$ decays with rate $O(k^{-\alpha})$ with $1 < \alpha < 2$ as $k \to \infty$. Thus, the approximation error using the first m terms measured in the L^2-norm, i.e., $\left\| f - \sum_{k=1}^{m} c_k \varphi_k \right\|_{L^2(\Omega)}$ should have a decay rate of $O\left(m^{-\alpha+0.5}\right)$ as $m \to \infty$.

The C^2-smoothness of the boundary could be weakened . . .

Conjecture on the Coefficient Decay Rate

Conjecture (NS 2007)

Let Ω be a C^2-domain of general shape and let $f \in C\left(\overline{\Omega}\right)$ with $\dfrac{\partial f}{\partial x_j} \in BV\left(\overline{\Omega}\right)$ for $j = 1, \ldots, d$. Let $\{c_k = \langle f, \varphi_k \rangle\}_{k \in \mathbb{N}}$ be the expansion coefficients of f with respect to our Laplacian eigenbasis on this domain. Then, $|c_k|$ decays with rate $O(k^{-\alpha})$ with $1 < \alpha < 2$ as $k \to \infty$. Thus, the approximation error using the first m terms measured in the L^2-norm, i.e., $\left\| f - \sum_{k=1}^{m} c_k \varphi_k \right\|_{L^2(\Omega)}$ should have a decay rate of $O\left(m^{-\alpha+0.5}\right)$ as $m \to \infty$.

The C^2-smoothness of the boundary could be weakened . . .

Outline

Hippocampal Shape Analysis

- Presenting the work of *Faisal Beg* and his group at Simon Fraser Univ. using our technique
- Want to distinguish people with mild dementia of the Alzheimer type (DAT) from cognitively normal (CN) people
- Hippocampus plays important roles in long-term memory and spatial navigation

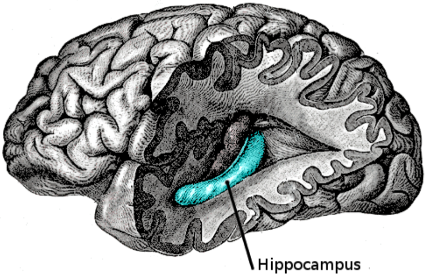


Figure: From Wikipedia

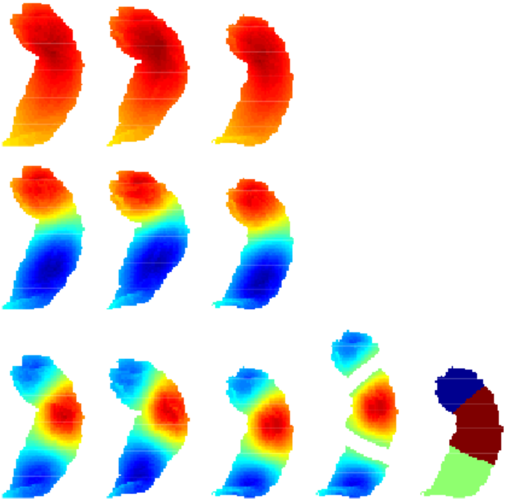
Hippocampal Shape Analysis . . .

- Dataset: Left hippocampus segmented from 3D MRI images
- Compute the smallest 999 Laplacian eigenvalues (i.e., the largest 999 eigenvalues of the integral operator $\mathscr{K}$) for each left hippocampus
- Construct a feature vector for each left hippocampus:

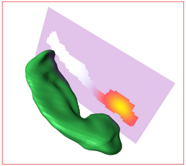
$$\boldsymbol{F} := \left(\frac{\lambda_1}{\lambda_2}, \ldots, \frac{\lambda_1}{\lambda_{n+1}}\right)^{\mathsf{T}} = \left(\frac{\mu_2}{\mu_1}, \ldots, \frac{\mu_{n+1}}{\mu_1}\right)^{\mathsf{T}} \in \mathbb{R}^n.$$

  This feature vector was used by Khabou, Hermi, and Rhouma (2007) for 2D shape classification (e.g., shapes of tree leaves).
- Reduce the feature space dimension via PCA to from $n = 998$ to n'
- Classified by the linear SVM (support vector machine)

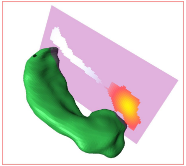
First Three Eigenfunctions of Three Patients



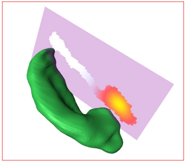
The Second Eigenfunction φ_2



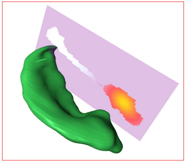
(a) $N = 15135$



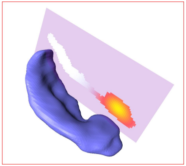
(b) $N = 15438$



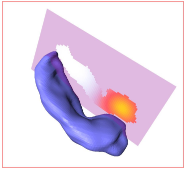
(c) $N = 14938$



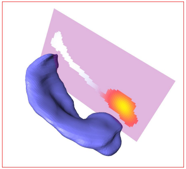
(d) $N = 15256$



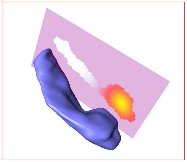
(e) $N = 14201$



(f) $N = 15630$

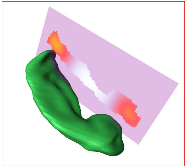


(g) $N = 12073$

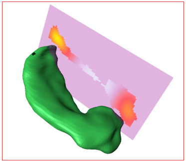


(h) $N = 12240$

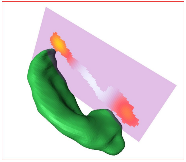
The Third Eigenfunction φ_3



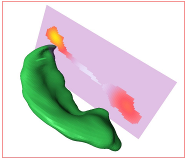
(a) $N = 15135$



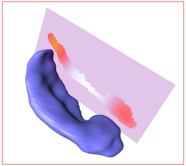
(b) $N = 15438$



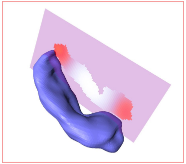
(c) $N = 14938$



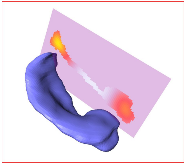
(d) $N = 15256$



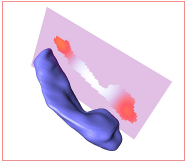
(e) $N = 14201$



(f) $N = 15630$



(g) $N = 12073$



(h) $N = 12240$

Classification Results

Dataset consists of the segmented left hippocampuses of 18 DAT subjects and of 26 CN subjects:

Method	Accuracy	Specificity	Sensitivity	n	n'
MomInv	68.1%	69.2%	66.6%	12	1
TensorInv	75.0%	76.9%	72.2%	$\geq 1.9E5$	17
LapEig	77.2%	84.6%	66.6%	998	14
GeodesicInv	86.3%	77.7%	92.3%	$\geq 1.3E6$	27

$$\text{accuracy} := \frac{|TP| + |TN|}{|\text{people examined}|} = \frac{|\text{people correctly diagnosed}|}{|\text{people examined}|}$$

$$\text{specificity} := \frac{|TN|}{|TN| + |FP|} = \frac{|\text{people correctly diagnosed as healthy}|}{|\text{healthy people examined}|}$$

$$\text{sensitivity} := \frac{|TP|}{|TP| + |FN|} = \frac{|\text{people correctly diagnosed as mild AD}|}{|\text{people with mild AD examined}|}$$

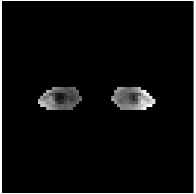
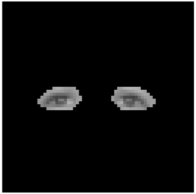
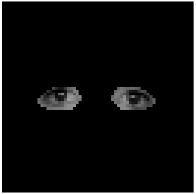
Outline

Comparison with PCA

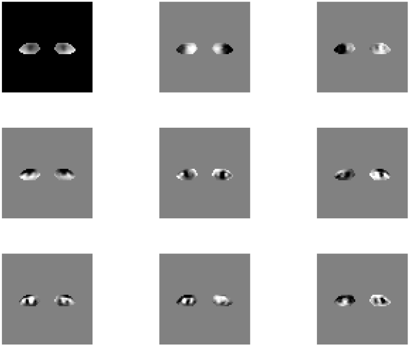
- Consider a stochastic process living on a domain Ω.
- *PCA/Karhunen-Loève Transform* is often used.
- PCA/KLT *implicitly* incorporate geometric information of the measurement (or pixel) location through *data correlation*.
- Our Laplacian eigenfunctions use *explicit* geometric information through the harmonic kernel $K(\boldsymbol{x}, \boldsymbol{y})$.

Comparison with PCA: Example

- *"Rogue's Gallery"* dataset from Larry Sirovich
- 72 training dataset; 71 test dataset
- Left & right eye regions

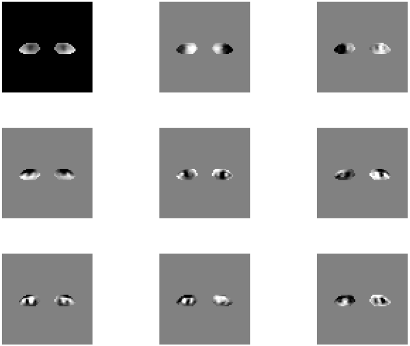




Comparison with PCA: Basis Vectors

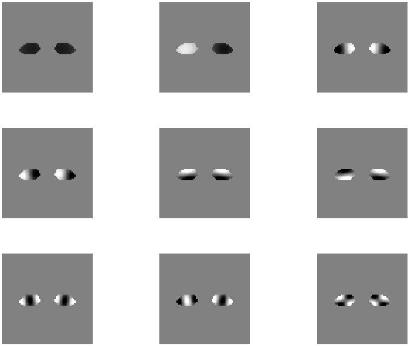


(a) KLB/PCA 1:9

Comparison with PCA: Basis Vectors

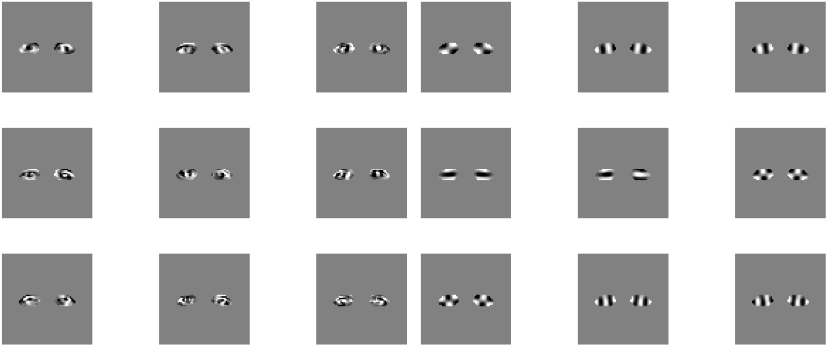


(a) KLB/PCA 1:9



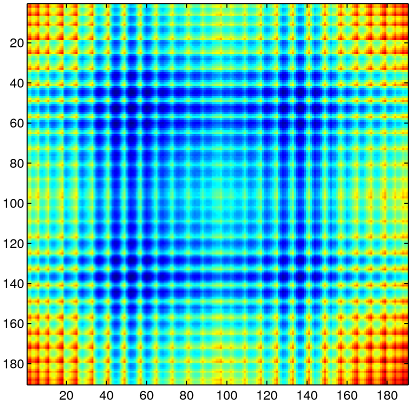
(b) Laplacian Eigenfunctions 1:9

Comparison with PCA: Basis Vectors . . .

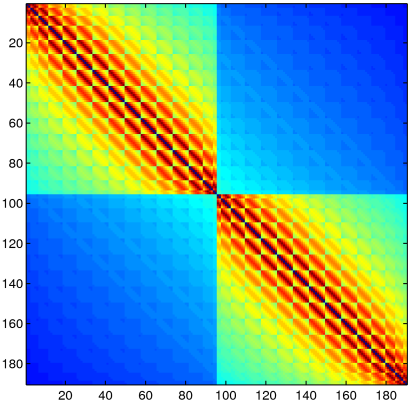


(a) KLB/PCA 10:18 (b) Laplacian Eigenfunctions 10:18

Comparison with PCA: Kernel Matrix

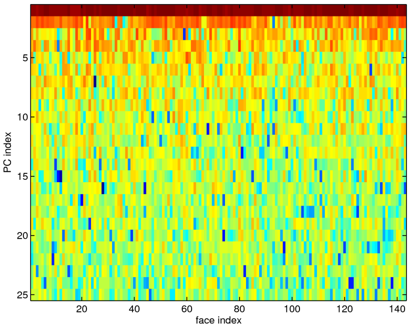


(a) Covariance

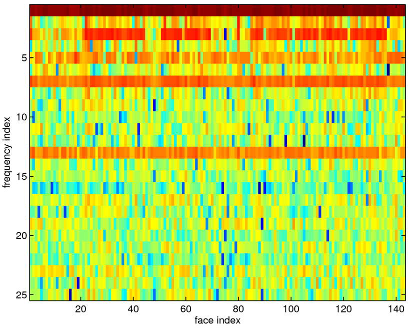


(b) Harmonic kernel

Comparison with PCA: Energy Distribution over Coordinates

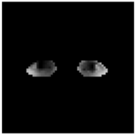


(a) KLB/PCA

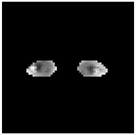


(b) Laplacian Eigenfunctions

Comparison with PCA: Basis Vector #7 . . .

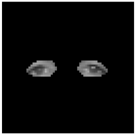


c_7:large

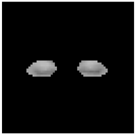


c_7:large

φ_7

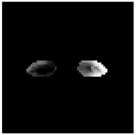


c_7:small

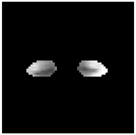


c_7:small

Comparison with PCA: Basis Vector #13 . . .



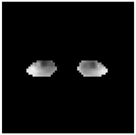
c_{13}:large



c_{13}:large

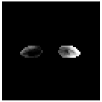
φ_{13}

c_{13}:small

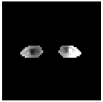


c_{13}:small

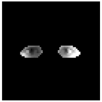
Asymmetry Detector



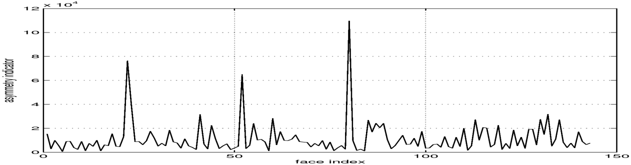
Eyes #80



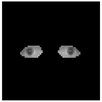
Eyes #22



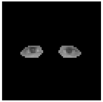
Eyes #52



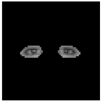
Asymmetry detector



Eyes #5



Eyes #84



Eyes #59

Outline

Introductory Remarks

- For much more details of this part of tutorial, please check my course website on "Harmonic Analysis on Graphs & Networks": `http://www.math.ucdavis.edu/~saito/courses/HarmGraph/`
- Good general references on the graph Laplacian *eigenvalues* are:
  - R. B. Bapat: *Graphs and Matrices*, Universitext, Springer, 2010.
  - F. R. K. Chung: *Spectral Graph Theory*, Amer. Math. Soc., 1997.
  - D. Cvetković, P. Rowlinson, & S. Simić: *An Introduction to the Theory of Graph Spectra*, Vol. 75, London Mathematical Society Student Texts, Cambridge Univ. Press, 2010.
- As for the graph Laplacian *eigenfunctions*, there are not too many books (although there may be many papers); one of the good books is
  - T. Bıyıkoğlu, J. Leydold, & P. F. Stadler, *Laplacian Eigenvectors of Graphs*, Lecture Notes in Mathematics, vol. 1915, Springer, 2007.
- As for *wavelet-like transforms* on graphs, there are many recent publications including those of my group. The following is a good survey paper:
  - D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, & P. Vandergheynst: "The emerging field of signal processing on graphs," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

Outline

Basic Definitions

- A graph G consists of a set of vertices (or nodes) V and a set of edges E connecting some pairs of vertices in V. We write $G = (V, E)$.
- An edge connecting a vertex $x \in V$ and itself is called a loop.
- For $x, y \in V$, if $\exists$ more than one edge connecting x and y, they are called multiple edges.
- A graph having loops or multiple edges is called a multiple graph (or multigraph); otherwise it is called a simple graph.

- In this tutorial, we shall only deal with simple graphs. So, when we say a graph, we mean a simple graph.

Basic Definitions

- A graph G consists of a set of vertices (or nodes) V and a set of edges E connecting some pairs of vertices in V. We write $G = (V, E)$.
- An edge connecting a vertex $x \in V$ and itself is called a loop.
- For $x, y \in V$, if $\exists$ more than one edge connecting x and y, they are called multiple edges.
- A graph having loops or multiple edges is called a multiple graph (or multigraph); otherwise it is called a simple graph.

- In this tutorial, we shall only deal with simple graphs. So, when we say a graph, we mean a simple graph.

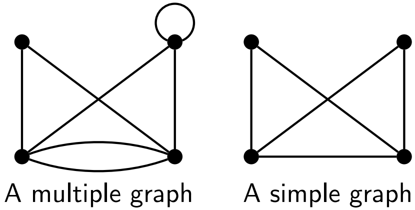
Basic Definitions

- A graph G consists of a set of vertices (or nodes) V and a set of edges E connecting some pairs of vertices in V. We write $G = (V, E)$.
- An edge connecting a vertex $x \in V$ and itself is called a loop.
- For $x, y \in V$, if $\exists$ more than one edge connecting x and y, they are called multiple edges.
- A graph having loops or multiple edges is called a multiple graph (or multigraph); otherwise it is called a simple graph.

- In this tutorial, we shall only deal with simple graphs. So, when we say a graph, we mean a simple graph.

Basic Definitions

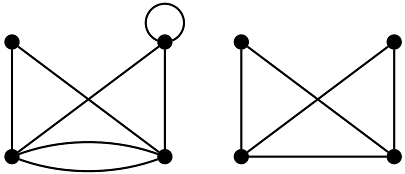
- A graph G consists of a set of vertices (or nodes) V and a set of edges E connecting some pairs of vertices in V. We write $G = (V, E)$.
- An edge connecting a vertex $x \in V$ and itself is called a loop.
- For $x, y \in V$, if $\exists$ more than one edge connecting x and y, they are called multiple edges.
- A graph having loops or multiple edges is called a multiple graph (or multigraph); otherwise it is called a simple graph.



- In this tutorial, we shall only deal with simple graphs. So, when we say a graph, we mean a simple graph.

Basic Definitions

- A graph G consists of a set of vertices (or nodes) V and a set of edges E connecting some pairs of vertices in V. We write $G = (V, E)$.
- An edge connecting a vertex $x \in V$ and itself is called a loop.
- For $x, y \in V$, if $\exists$ more than one edge connecting x and y, they are called multiple edges.
- A graph having loops or multiple edges is called a multiple graph (or multigraph); otherwise it is called a simple graph.



A multiple graph   A simple graph

- In this tutorial, we shall only deal with simple graphs. So, when we say a graph, we mean a simple graph.

- If two distinct vertices $x, y \in V$ are connected by an edge e, then x, y are called the endpoints (or ends) of e, and x, y are said to be adjacent, and we write $x \sim y$. We also say an edge e is incident with x and y, and e joins x and y.
- The number of edges that are incident with x (i.e., have x as their endpoint) = the degree (or valency) of x and write $d(x)$ or d_x.
- If the number of vertices $|V| < \infty$, then G is called a finite graph; otherwise an infinite graph.
- If each edge in E has a direction, G is called a directed graph or digraph, and such E is written as $\boldsymbol{E}$.

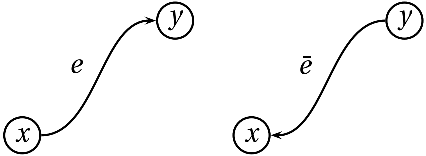
- If $e = [x, y]$, then x and y are called a tail and a head, respectively.

- If two distinct vertices $x, y \in V$ are connected by an edge e, then x, y are called the endpoints (or ends) of e, and x, y are said to be adjacent, and we write $x \sim y$. We also say an edge e is incident with x and y, and e joins x and y.
- The number of edges that are incident with x (i.e., have x as their endpoint) = the degree (or valency) of x and write $d(x)$ or d_x.
- If the number of vertices $|V| < \infty$, then G is called a finite graph; otherwise an infinite graph.
- If each edge in E has a direction, G is called a directed graph or digraph, and such E is written as $\mathbf{E}$.
- If $e = [x, y]$, then x and y are called a tail and a head, respectively.

- If two distinct vertices $x, y \in V$ are connected by an edge e, then x, y are called the endpoints (or ends) of e, and x, y are said to be adjacent, and we write $x \sim y$. We also say an edge e is incident with x and y, and e joins x and y.
- The number of edges that are incident with x (i.e., have x as their endpoint) = the degree (or valency) of x and write $d(x)$ or d_x.
- If the number of vertices $|V| < \infty$, then G is called a finite graph; otherwise an infinite graph.
- If each edge in E has a direction, G is called a directed graph or digraph, and such E is written as $\boldsymbol{E}$.

- If $e = [x, y]$, then x and y are called a tail and a head, respectively.

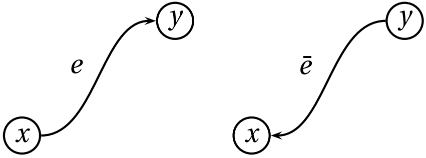
- If two distinct vertices $x, y \in V$ are connected by an edge e, then x, y are called the endpoints (or ends) of e, and x, y are said to be adjacent, and we write $x \sim y$. We also say an edge e is incident with x and y, and e joins x and y.
- The number of edges that are incident with x (i.e., have x as their endpoint) = the degree (or valency) of x and write $d(x)$ or d_x.
- If the number of vertices $|V| < \infty$, then G is called a finite graph; otherwise an infinite graph.
- If each edge in E has a direction, G is called a directed graph or digraph, and such E is written as $\boldsymbol{E}$.



$e = [x, y]$ $\qquad$ $\bar{e} = [y, x]$

- If $e = [x, y]$, then x and y are called a tail and a head, respectively.

- If two distinct vertices $x, y \in V$ are connected by an edge e, then x, y are called the endpoints (or ends) of e, and x, y are said to be adjacent, and we write $x \sim y$. We also say an edge e is incident with x and y, and e joins x and y.
- The number of edges that are incident with x (i.e., have x as their endpoint) = the degree (or valency) of x and write $d(x)$ or d_x.
- If the number of vertices $|V| < \infty$, then G is called a finite graph; otherwise an infinite graph.
- If each edge in E has a direction, G is called a directed graph or digraph, and such E is written as $\boldsymbol{E}$.



$e = [x, y]$ $\qquad$ $\bar{e} = [y, x]$

- If $e = [x, y]$, then x and y are called a tail and a head, respectively.

- If an edge e does not have a direction, we write $e = (x, y)$.
- If each edge $e = (x, y)$ of G has a weight (normally positive), written as $w_e = w_{xy}$, then G is called a weighted graph. G is said to be unweighted if $w_e = \text{const.}$ for each $e \in E$, and normally w_e is set to 1.
- A path from x to y in a graph G is a subgraph of G consisting of a sequence of distinct vertices starting with x and ending with y such that consecutive vertices are adjacent. A path starting from x that returns to x (but is not a loop) is called a cycle.
- For any two vertices in V, if $\exists$ a path connecting them, then such a graph G is said to be connected. In the case of a digraph, it is said to be strongly connected.
- A tree is a connected graph without cycles, and is often denoted by T instead of G. For a tree T, we have $|E(T)| = |V(T)| - 1$, where $|\cdot|$ denotes a cardinality of a set.
- The length (or cost) $\ell(P)$ of a path P is the sum of its corresponding edge weights, i.e., $\ell(P) := \sum_{e \in E(P)} w_e$. Let $\mathscr{P}_{xy}$ be a set of all possible paths from x to y in G. The graph distance from x to y is defined by $d(x, y) := \inf_{P \in \mathscr{P}_{xy}} \ell(P)$.

- If an edge e does not have a direction, we write $e = (x, y)$.
- If each edge $e = (x, y)$ of G has a weight (normally positive), written as $w_e = w_{xy}$, then G is called a weighted graph. G is said to be unweighted if $w_e = \text{const.}$ for each $e \in E$, and normally w_e is set to 1.
- A path from x to y in a graph G is a subgraph of G consisting of a sequence of distinct vertices starting with x and ending with y such that consecutive vertices are adjacent. A path starting from x that returns to x (but is not a loop) is called a cycle.
- For any two vertices in V, if $\exists$ a path connecting them, then such a graph G is said to be connected. In the case of a digraph, it is said to be strongly connected.
- A tree is a connected graph without cycles, and is often denoted by T instead of G. For a tree T, we have $|E(T)| = |V(T)| - 1$, where $|\cdot|$ denotes a cardinality of a set.
- The length (or cost) $\ell(P)$ of a path P is the sum of its corresponding edge weights, i.e., $\ell(P) := \sum_{e \in E(P)} w_e$. Let $\mathscr{P}_{xy}$ be a set of all possible paths from x to y in G. The graph distance from x to y is defined by $d(x, y) := \inf_{P \in \mathscr{P}_{xy}} \ell(P)$.

- If an edge e does not have a direction, we write $e = (x, y)$.
- If each edge $e = (x, y)$ of G has a weight (normally positive), written as $w_e = w_{xy}$, then G is called a weighted graph. G is said to be unweighted if $w_e = \mathrm{const.}$ for each $e \in E$, and normally w_e is set to 1.
- A path from x to y in a graph G is a subgraph of G consisting of a sequence of distinct vertices starting with x and ending with y such that consecutive vertices are adjacent. A path starting from x that returns to x (but is not a loop) is called a cycle.
- For any two vertices in V, if $\exists$ a path connecting them, then such a graph G is said to be connected. In the case of a digraph, it is said to be strongly connected.
- A tree is a connected graph without cycles, and is often denoted by T instead of G. For a tree T, we have $|E(T)| = |V(T)| - 1$, where $|\cdot|$ denotes a cardinality of a set.
- The length (or cost) $\ell(P)$ of a path P is the sum of its corresponding edge weights, i.e., $\ell(P) := \sum_{e \in E(P)} w_e$. Let $\mathscr{P}_{xy}$ be a set of all possible paths from x to y in G. The graph distance from x to y is defined by $d(x, y) := \inf_{P \in \mathscr{P}_{xy}} \ell(P)$.

- If an edge e does not have a direction, we write $e = (x, y)$.
- If each edge $e = (x, y)$ of G has a weight (normally positive), written as $w_e = w_{xy}$, then G is called a weighted graph. G is said to be unweighted if $w_e = \text{const.}$ for each $e \in E$, and normally w_e is set to 1.
- A path from x to y in a graph G is a subgraph of G consisting of a sequence of distinct vertices starting with x and ending with y such that consecutive vertices are adjacent. A path starting from x that returns to x (but is not a loop) is called a cycle.
- For any two vertices in V, if $\exists$ a path connecting them, then such a graph G is said to be connected. In the case of a digraph, it is said to be strongly connected.
- A tree is a connected graph without cycles, and is often denoted by T instead of G. For a tree T, we have $|E(T)| = |V(T)| - 1$, where $|\cdot|$ denotes a cardinality of a set.
- The length (or cost) $\ell(P)$ of a path P is the sum of its corresponding edge weights, i.e., $\ell(P) := \sum_{e \in E(P)} w_e$. Let $\mathscr{P}_{xy}$ be a set of all possible paths from x to y in G. The graph distance from x to y is defined by $d(x, y) := \inf_{P \in \mathscr{P}_{xy}} \ell(P)$.

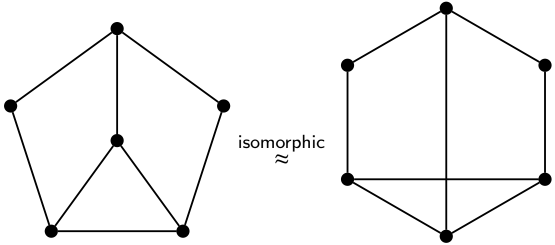
- If an edge e does not have a direction, we write $e = (x, y)$.
- If each edge $e = (x, y)$ of G has a weight (normally positive), written as $w_e = w_{xy}$, then G is called a weighted graph. G is said to be unweighted if $w_e = \text{const.}$ for each $e \in E$, and normally w_e is set to 1.
- A path from x to y in a graph G is a subgraph of G consisting of a sequence of distinct vertices starting with x and ending with y such that consecutive vertices are adjacent. A path starting from x that returns to x (but is not a loop) is called a cycle.
- For any two vertices in V, if $\exists$ a path connecting them, then such a graph G is said to be connected. In the case of a digraph, it is said to be strongly connected.
- A tree is a connected graph without cycles, and is often denoted by T instead of G. For a tree T, we have $|E(T)| = |V(T)| - 1$, where $|\cdot|$ denotes a cardinality of a set.
- The length (or cost) $\ell(P)$ of a path P is the sum of its corresponding edge weights, i.e., $\ell(P) := \sum_{e \in E(P)} w_e$. Let $\mathscr{P}_{xy}$ be a set of all possible paths from x to y in G. The graph distance from x to y is defined by $d(x, y) := \inf_{P \in \mathscr{P}_{xy}} \ell(P)$.

- If an edge e does not have a direction, we write $e = (x, y)$.
- If each edge $e = (x, y)$ of G has a weight (normally positive), written as $w_e = w_{xy}$, then G is called a weighted graph. G is said to be unweighted if $w_e = \text{const.}$ for each $e \in E$, and normally w_e is set to 1.
- A path from x to y in a graph G is a subgraph of G consisting of a sequence of distinct vertices starting with x and ending with y such that consecutive vertices are adjacent. A path starting from x that returns to x (but is not a loop) is called a cycle.
- For any two vertices in V, if $\exists$ a path connecting them, then such a graph G is said to be connected. In the case of a digraph, it is said to be strongly connected.
- A tree is a connected graph without cycles, and is often denoted by T instead of G. For a tree T, we have $|E(T)| = |V(T)| - 1$, where $|\cdot|$ denotes a cardinality of a set.
- The length (or cost) $\ell(P)$ of a path P is the sum of its corresponding edge weights, i.e., $\ell(P) := \sum_{e \in E(P)} w_e$. Let $\mathscr{P}_{xy}$ be a set of all possible paths from x to y in G. The graph distance from x to y is defined by $d(x, y) := \inf_{P \in \mathscr{P}_{xy}} \ell(P)$.

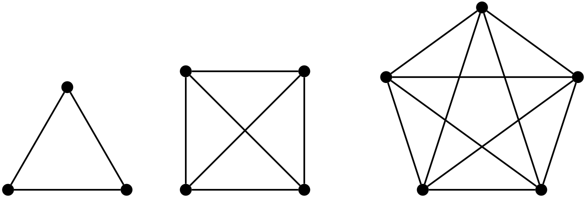
- Clearly, for an undirected graph, we always have $d(x,y) = d(y,x)$, but that is not the case for a directed graph in general.
- $\mathrm{diam}(G) := \sup_{x,y \in V} d(x,y)$ is called the diameter of G. Note that $\mathrm{diam}(G) < \infty \iff G$ is finite.
- We say two graphs are isomorphic if $\exists$ a one-to-one correspondence between the vertex sets such that if two vertices are joined by an edge in one graph, the corresponding vertices are also joined by an edge in the other graph.

- Clearly, for an undirected graph, we always have $d(x, y) = d(y, x)$, but that is not the case for a directed graph in general.
- $\mathrm{diam}(G) := \sup_{x,y \in V} d(x, y)$ is called the diameter of G. Note that $\mathrm{diam}(G) < \infty \Longleftrightarrow G$ is finite.
- We say two graphs are isomorphic if $\exists$ a one-to-one correspondence between the vertex sets such that if two vertices are joined by an edge in one graph, the corresponding vertices are also joined by an edge in the other graph.

- Clearly, for an undirected graph, we always have $d(x,y) = d(y,x)$, but that is not the case for a directed graph in general.
- $\operatorname{diam}(G) := \sup_{x,y \in V} d(x,y)$ is called the diameter of G. Note that $\operatorname{diam}(G) < \infty \iff G$ is finite.
- We say two graphs are isomorphic if $\exists$ a one-to-one correspondence between the vertex sets such that if two vertices are joined by an edge in one graph, the corresponding vertices are also joined by an edge in the other graph.



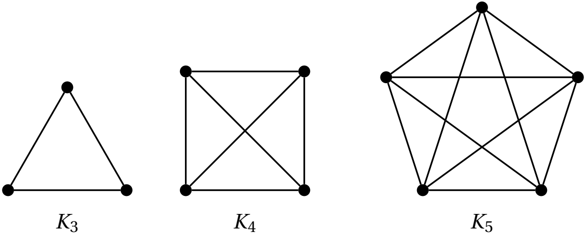
- The complete graph K_n on n vertices is a simple graph that has all possible $\binom{n}{2}$ edges.



K_3 K_4 K_5

- If all the vertices of a graph has the same degree, the graph is called regular. Hence, K_n is regular.

- The complete graph K_n on n vertices is a simple graph that has all possible $\binom{n}{2}$ edges.



- If all the vertices of a graph has the same degree, the graph is called regular. Hence, K_n is regular.

Matrices Associated with a Graph

- The adjacency matrix $A = A(G) = (a_{ij}) \in \mathbb{R}^{n \times n}$, $n = |V|$, for an unweighted graph G consists of the following entries:

$$a_{ij} := \begin{cases} 1 & \text{if } v_i \sim v_j; \\ 0 & \text{otherwise.} \end{cases}$$

- Another typical way to define its entries is based on the similarity of information at v_i and v_j:

$$a_{ij} := \exp(-\mathrm{dist}(v_i, v_j)^2 / \epsilon^2)$$

where dist is an appropriate distance measure (i.e., metric) defined in V, and $\epsilon > 0$ is an appropriate scale parameter. This leads to a weighted graph. We will discuss later more about the weighted graphs, how to determine weights, and how to construct a graph from given datasets in general.

Matrices Associated with a Graph

- The adjacency matrix $A = A(G) = (a_{ij}) \in \mathbb{R}^{n \times n}$, $n = |V|$, for an unweighted graph G consists of the following entries:

$$a_{ij} := \begin{cases} 1 & \text{if } v_i \sim v_j; \\ 0 & \text{otherwise.} \end{cases}$$

- Another typical way to define its entries is based on the similarity of information at v_i and v_j:

$$a_{ij} := \exp(-\mathrm{dist}(v_i, v_j)^2 / \epsilon^2)$$

  where dist is an appropriate distance measure (i.e., metric) defined in V, and $\epsilon > 0$ is an appropriate scale parameter. This leads to a weighted graph. We will discuss later more about the weighted graphs, how to determine weights, and how to construct a graph from given datasets in general.

Matrices Associated with a Graph ...

- The degree matrix $D = D(G) = \mathrm{diag}(d_1, \ldots, d_n) \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose entries are:

$$d_i = d(v_i) = d_{v_i} := \sum_{j=1}^{n} a_{ij}.$$

Note that the above definition works for both unweighted and weighted graphs.

- The transition matrix $P = P(G) = (p_{ij}) \in \mathbb{R}^{n \times n}$ consists of the following entries:

$$p_{ij} := a_{ij}/d_i \quad \text{if } d_i \neq 0.$$

- p_{ij} represents the probability of a random walk from v_i to v_j in one step: $\sum_j p_{ij} = 1$, i.e., P is row stochastic.
- $A^{\mathsf{T}} = A$, $P^{\mathsf{T}} \neq P$, $P = D^{-1}A$.

Matrices Associated with a Graph . . .

- The degree matrix $D = D(G) = \mathrm{diag}(d_1, \ldots, d_n) \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose entries are:

$$d_i = d(v_i) = d_{v_i} := \sum_{j=1}^{n} a_{ij}.$$

  Note that the above definition works for both unweighted and weighted graphs.

- The transition matrix $P = P(G) = (p_{ij}) \in \mathbb{R}^{n\times n}$ consists of the following entries:

$$p_{ij} := a_{ij}/d_i \quad \text{if } d_i \neq 0.$$

- p_{ij} represents the probability of a random walk from v_i to v_j in one step: $\sum_j p_{ij} = 1$, i.e., P is row stochastic.
- $A^{\mathsf{T}} = A$, $P^{\mathsf{T}} \neq P$, $P = D^{-1}A$.

Matrices Associated with a Graph ...

- The degree matrix $D = D(G) = \mathrm{diag}(d_1, \ldots, d_n) \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose entries are:

$$d_i = d(v_i) = d_{v_i} := \sum_{j=1}^{n} a_{ij}.$$

Note that the above definition works for both unweighted and weighted graphs.

- The transition matrix $P = P(G) = (p_{ij}) \in \mathbb{R}^{n\times n}$ consists of the following entries:

$$p_{ij} := a_{ij}/d_i \quad \text{if } d_i \neq 0.$$

- p_{ij} represents the probability of a random walk from v_i to v_j in one step: $\sum_j p_{ij} = 1$, i.e., P is row stochastic.
- $A^{\mathsf{T}} = A$, $P^{\mathsf{T}} \neq P$, $P = D^{-1}A$.

Matrices Associated with a Graph ...

- The degree matrix $D = D(G) = \text{diag}(d_1, \dots, d_n) \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose entries are:

$$d_i = d(v_i) = d_{v_i} := \sum_{j=1}^{n} a_{ij}.$$

  Note that the above definition works for both unweighted and weighted graphs.

- The transition matrix $P = P(G) = (p_{ij}) \in \mathbb{R}^{n \times n}$ consists of the following entries:

$$p_{ij} := a_{ij} / d_i \quad \text{if } d_i \neq 0.$$

- p_{ij} represents the probability of a random walk from v_i to v_j in one step: $\sum_j p_{ij} = 1$, i.e., P is row stochastic.
- $A^\mathsf{T} = A$, $P^\mathsf{T} \neq P$, $P = D^{-1} A$.

Matrices Associated with a Graph . . .

- Let G be an *undirected* graph. Then, we can define several Laplacian matrices of G:

$$L(G) := D - A \qquad \text{Unnormalized}$$

$$L_{\mathrm{rw}}(G) := I_n - D^{-1}A = I_n - P = D^{-1}L \qquad \text{Normalized}$$

$$L_{\mathrm{sym}}(G) := I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} \qquad \text{Symmetrically-Normalized}$$

- The signless Laplacian is defined as follows, but we will not deal with this in this tutorial: $Q(G) := D + A$.
- Graph Laplacians can also be defined for directed graphs; However, there are many different definitions based on the types/classes of directed graphs, and in general, those matrices are *nonsymmetric*. See, e.g., Fan Chung: "Laplacians and the Cheeger inequality for directed graphs," *Ann. Comb.*, vol. 9, no. 1, pp. 1–19, 2005, for an attempt to symmetrize graph Laplacian matrices for *strongly connected* digraphs.

Matrices Associated with a Graph . . .

- Let G be an *undirected* graph. Then, we can define several Laplacian matrices of G:

$$L(G) := D - A \qquad \text{Unnormalized}$$

$$L_{\mathrm{rw}}(G) := I_n - D^{-1}A = I_n - P = D^{-1}L \qquad \text{Normalized}$$

$$L_{\mathrm{sym}}(G) := I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} \qquad \text{Symmetrically-Normalized}$$

- The signless Laplacian is defined as follows, but we will not deal with this in this tutorial: $Q(G) := D + A$.
- Graph Laplacians can also be defined for directed graphs; However, there are many different definitions based on the types/classes of directed graphs, and in general, those matrices are *nonsymmetric*. See, e.g., Fan Chung: "Laplacians and the Cheeger inequality for directed graphs," *Ann. Comb.*, vol. 9, no. 1, pp. 1–19, 2005, for an attempt to symmetrize graph Laplacian matrices for *strongly connected* digraphs.

Matrices Associated with a Graph ...

- Let G be an *undirected* graph. Then, we can define several Laplacian matrices of G:

$$L(G) := D - A \qquad \text{Unnormalized}$$

$$L_{\mathrm{rw}}(G) := I_n - D^{-1}A = I_n - P = D^{-1}L \qquad \text{Normalized}$$

$$L_{\mathrm{sym}}(G) := I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} \qquad \text{Symmetrically-Normalized}$$

- The signless Laplacian is defined as follows, but we will not deal with this in this tutorial: $Q(G) := D + A$.
- Graph Laplacians can also be defined for directed graphs; However, there are many different definitions based on the types/classes of directed graphs, and in general, those matrices are *nonsymmetric*. See, e.g., Fan Chung: "Laplacians and the Cheeger inequality for directed graphs," *Ann. Comb.*, vol. 9, no. 1, pp. 1–19, 2005, for an attempt to symmetrize graph Laplacian matrices for *strongly connected* digraphs.

Functions Defined on a Graph

$$C(V) := \{\text{all functions defined on } V\}$$

$$C_0(V) := \{f \in C(V) \mid \operatorname{supp} f \text{ is a finite subset of } V\}$$

$$\operatorname{supp} f := \{u \in V \mid f(u) \neq 0\}$$

$$\langle f, g \rangle := \sum_{u \in V} f(u) g(u)$$

$$\langle f, g \rangle_{\#} := \sum_{u \in V} d(u) f(u) g(u)$$

$$\mathcal{L}^2(V) := \left\{ f \in C(V) \,\middle|\, \|f\|_{\#} := \sqrt{\langle f, f \rangle_{\#}} < \infty \right\}$$

Lemma

$$\langle Pf, g \rangle_{\#} = \langle f, Pg \rangle_{\#} \quad \forall f, g \in \mathcal{L}^2(V);$$

$$\|Pf\|_{\#} \leq \|f\|_{\#} \quad \forall f \in \mathcal{L}^2(V).$$

Functions Defined on a Graph . . .

- Let $f \in \mathcal{L}^2(V)$. Then

$$Lf(v_i) = d_i f(v_i) - \sum_{j=1}^{n} a_{ij} f(v_j) = \sum_{j=1}^{n} a_{ij} \left(f(v_i) - f(v_j) \right).$$

  i.e., this is a generalization of the *finite difference approximation* to the Laplace operator.

- On the other hand,

$$L_{\mathrm{rw}} f(v_i) = f(v_i) - \sum_{j=1}^{n} p_{ij} f(v_j) = \frac{1}{d_i} \sum_{j=1}^{n} a_{ij} \left(f(v_i) - f(v_j) \right).$$

$$L_{\mathrm{sym}} f(v_i) = f(v_i) - \frac{1}{\sqrt{d_i}} \sum_{j=1}^{n} \frac{a_{ij}}{\sqrt{d_j}} f(v_j) = \frac{1}{\sqrt{d_i}} \sum_{j=1}^{n} a_{ij} \left(\frac{f(v_i)}{\sqrt{d_i}} - \frac{f(v_j)}{\sqrt{d_j}} \right).$$

- Note that these definitions of the graph Laplacian corresponds to $-\Delta$ in $\mathbb{R}^d$, i.e., they are nonnegative operators (a.k.a. positive semi-definite matrices).

Functions Defined on a Graph . . .

- Let $f \in \mathcal{L}^2(V)$. Then

$$Lf(v_i) = d_i f(v_i) - \sum_{j=1}^{n} a_{ij} f(v_j) = \sum_{j=1}^{n} a_{ij} \left(f(v_i) - f(v_j) \right).$$

  i.e., this is a generalization of the *finite difference approximation* to the Laplace operator.

- On the other hand,

$$L_{\mathrm{rw}} f(v_i) = f(v_i) - \sum_{j=1}^{n} p_{ij} f(v_j) = \frac{1}{d_i} \sum_{j=1}^{n} a_{ij} \left(f(v_i) - f(v_j) \right).$$

$$L_{\mathrm{sym}} f(v_i) = f(v_i) - \frac{1}{\sqrt{d_i}} \sum_{j=1}^{n} \frac{a_{ij}}{\sqrt{d_j}} f(v_j) = \frac{1}{\sqrt{d_i}} \sum_{j=1}^{n} a_{ij} \left(\frac{f(v_i)}{\sqrt{d_i}} - \frac{f(v_j)}{\sqrt{d_j}} \right).$$

- Note that these definitions of the graph Laplacian corresponds to $-\Delta$ in $\mathbb{R}^d$, i.e., they are nonnegative operators (a.k.a. positive semi-definite matrices).

Functions Defined on a Graph ...

- Let $f \in \mathcal{L}^2(V)$. Then

$$Lf(v_i) = d_i f(v_i) - \sum_{j=1}^{n} a_{ij} f(v_j) = \sum_{j=1}^{n} a_{ij} \left(f(v_i) - f(v_j) \right).$$

  i.e., this is a generalization of the *finite difference approximation* to the Laplace operator.

- On the other hand,

$$L_{\mathrm{rw}} f(v_i) = f(v_i) - \sum_{j=1}^{n} p_{ij} f(v_j) = \frac{1}{d_i} \sum_{j=1}^{n} a_{ij} \left(f(v_i) - f(v_j) \right).$$

$$L_{\mathrm{sym}} f(v_i) = f(v_i) - \frac{1}{\sqrt{d_i}} \sum_{j=1}^{n} \frac{a_{ij}}{\sqrt{d_j}} f(v_j) = \frac{1}{\sqrt{d_i}} \sum_{j=1}^{n} a_{ij} \left(\frac{f(v_i)}{\sqrt{d_i}} - \frac{f(v_j)}{\sqrt{d_j}} \right).$$

- Note that these definitions of the graph Laplacian corresponds to $-\Delta$ in $\mathbb{R}^d$, i.e., they are nonnegative operators (a.k.a. positive semi-definite matrices).

Functions Defined on a Graph . . .

- A function $f \in C(V)$ is called harmonic if

$$Lf = 0, \; L_{\mathrm{rw}} f = 0, \text{ or } \; L_{\mathrm{sym}} f = 0.$$

- A function $f \in C(V)$ is called superharmonic at $x \in V$ if

$$Lf(x) \geq 0, \; L_{\mathrm{rw}} f(x) \geq 0, \text{ or } \; L_{\mathrm{sym}} f(x) \geq 0.$$

- These corresponds to:

$$f(v_i) \geq \frac{1}{d_i} \sum_{j=1}^{n} a_{ij} f(v_j), \; f(v_i) \geq \sum_{j=1}^{n} p_{ij} f(v_j), \text{ or } \; f(v_i) \geq \sum_{j=1}^{n} \frac{a_{ij}}{\sqrt{d_i}\sqrt{d_j}} f(v_j).$$

- One can also generalize various analytic concepts such as Green's functions, Green's identity, analytic functions, Cauchy-Riemann equations, . . . , to the graph setting!

Functions Defined on a Graph . . .

- A function $f \in C(V)$ is called harmonic if

$$Lf = 0, \; L_{\mathrm{rw}} f = 0, \text{ or } \; L_{\mathrm{sym}} f = 0.$$

- A function $f \in C(V)$ is called superharmonic at $x \in V$ if

$$Lf(x) \geq 0, \; L_{\mathrm{rw}} f(x) \geq 0, \text{ or } \; L_{\mathrm{sym}} f(x) \geq 0.$$

- These corresponds to:

$$f(v_i) \geq \frac{1}{d_i} \sum_{j=1}^{n} a_{ij} f(v_j), \; f(v_i) \geq \sum_{j=1}^{n} p_{ij} f(v_j), \text{ or } \; f(v_i) \geq \sum_{j=1}^{n} \frac{a_{ij}}{\sqrt{d_i}\sqrt{d_j}} f(v_j).$$

- One can also generalize various analytic concepts such as Green's functions, Green's identity, analytic functions, Cauchy-Riemann equations, . . . , to the graph setting!

Functions Defined on a Graph . . .

- A function $f \in C(V)$ is called harmonic if

$$Lf = 0,\ L_{\mathrm{rw}} f = 0, \text{ or } \ L_{\mathrm{sym}} f = 0.$$

- A function $f \in C(V)$ is called superharmonic at $x \in V$ if

$$Lf(x) \geq 0,\ L_{\mathrm{rw}} f(x) \geq 0, \text{ or } \ L_{\mathrm{sym}} f(x) \geq 0.$$

- These corresponds to:

$$f(v_i) \geq \frac{1}{d_i} \sum_{j=1}^{n} a_{ij} f(v_j),\ f(v_i) \geq \sum_{j=1}^{n} p_{ij} f(v_j), \text{ or } \ f(v_i) \geq \sum_{j=1}^{n} \frac{a_{ij}}{\sqrt{d_i}\sqrt{d_j}} f(v_j).$$

- One can also generalize various analytic concepts such as Green's functions, Green's identity, analytic functions, Cauchy-Riemann equations, . . . , to the graph setting!

Functions Defined on a Graph . . .

- A function $f \in C(V)$ is called harmonic if

$$Lf = 0, \; L_{\mathrm{rw}} f = 0, \text{ or } \; L_{\mathrm{sym}} f = 0.$$

- A function $f \in C(V)$ is called superharmonic at $x \in V$ if

$$Lf(x) \geq 0, \; L_{\mathrm{rw}} f(x) \geq 0, \text{ or } \; L_{\mathrm{sym}} f(x) \geq 0.$$

- These corresponds to:

$$f(v_i) \geq \frac{1}{d_i} \sum_{j=1}^{n} a_{ij} f(v_j), \; f(v_i) \geq \sum_{j=1}^{n} p_{ij} f(v_j), \text{ or } \; f(v_i) \geq \sum_{j=1}^{n} \frac{a_{ij}}{\sqrt{d_i}\sqrt{d_j}} f(v_j).$$

- One can also generalize various analytic concepts such as Green's functions, Green's identity, analytic functions, Cauchy-Riemann equations, . . . , to the graph setting!

Derivatives and Green's Identity

Let $C(\boldsymbol{E}) := \{\varphi$ defined on $\boldsymbol{E} \mid \varphi(\bar{e}) = -\varphi(e), e \in \boldsymbol{E}\}$. For $f \in C(V)$, define the derivative $df \in C(\boldsymbol{E})$ of f as

$$df(e) = df([x,y]) := f(y) - f(x).$$

Derivatives and Green's Identity

Let $C(\boldsymbol{E}) := \{\varphi \text{ defined on } \boldsymbol{E} \mid \varphi(\bar{e}) = -\varphi(e), e \in \boldsymbol{E}\}$. For $f \in C(V)$, define the derivative $df \in C(\boldsymbol{E})$ of f as

$$df(e) = df([x, y]) := f(y) - f(x).$$

Theorem (The discrete version of Green's first identity, Dodziuk 1984)

$$\forall f_1, f_2 \in C_0(V),\ \langle df_1, df_2 \rangle = \langle L_{\mathrm{rw}} f_1, f_2 \rangle_{\#} = \langle L f_1, f_2 \rangle$$

Derivatives and Green's Identity

Let $C(\boldsymbol{E}) := \{\varphi \text{ defined on } \boldsymbol{E} \mid \varphi(\bar{e}) = -\varphi(e), e \in \boldsymbol{E}\}$. For $f \in C(V)$, define the derivative $df \in C(\boldsymbol{E})$ of f as

$$df(e) = df([x,y]) := f(y) - f(x).$$

Theorem (The discrete version of Green's first identity, Dodziuk 1984)

$$\forall f_1, f_2 \in C_0(V), \langle df_1, df_2 \rangle = \langle L_{\mathrm{rw}} f_1, f_2 \rangle_{\#} = \langle L f_1, f_2 \rangle$$

Corollary

L, L_{rw}, and L_{sym} are nonnegative operators, e.g.,

$$\langle L_{\mathrm{rw}} f, f \rangle_{\#} = \langle Lf, f \rangle = \langle df, df \rangle \geq 0.$$

The Minimum Principle

Theorem (The discrete version of the minimum principle)

Let $f \in C(V)$ *be superharmonic at* $x \in V$. *If* $f(x) \le \min_{y \sim x} f(y)$, *then* $f(z) = f(x), \forall z \sim x$.

The Minimum Principle

Theorem (The discrete version of the minimum principle)

Let $f \in C(V)$ be superharmonic at $x \in V$. If $f(x) \le \min_{y\sim x} f(y)$, then $f(z) = f(x)$, $\forall z \sim x$.

Proof. From the superharmonicity of f at $x \in V$, we have

$$\frac{1}{d_x}\sum_{y\sim x} a_{xy} f(y) \le f(x).$$

On the other hand, from the condition of this theorem, we have

$$\frac{1}{d_x}\sum_{y\sim x} a_{xy} f(y) \ge \frac{1}{d_x}\sum_{y\sim x} a_{xy} f(x) = f(x).$$

Hence, we must have $\frac{1}{d_x}\sum_{y\sim x} a_{xy} f(y) = f(x)$. But this can happen only if $f(z) = f(x)$, $\forall z \sim x$. □

Why Graph Laplacians?

- We already know that the Laplacian eigenvalues and eigenfunctions are extremely useful for general domains in $\mathbb{R}^d$.
- The graph Laplacian *eigenvalues* reflect various intrinsic geometric and topological information about the graph including connectivity or the number of separated components; diameter; mean distance, . . .
- Fan Chung: *Spectral Graph Theory*, Amer. Math. Soc., 1997, says: *"This monograph is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs."*
- Due to the time limitation, I will *not* be able to discuss the details on how the graph Laplacian *eigenvalues* reveal the geometric and topological information of the graph. For the details, please check the above book, the books listed in the beginning of this section, and
  - R. Merris: "Laplacian matrices of graphs: a survey," *Linear Algebra Appl.*, vol. 197/198, pp. 143–176, 1994.
  - N. Saito & E. Woei: "Analysis of neuronal dendrite patterns using eigenvalues of graph Laplacians," *Japan SIAM Lett.* vol. 1 pp. 13–16, 2009 (Invited paper).

Why Graph Laplacians?

- We already know that the Laplacian eigenvalues and eigenfunctions are extremely useful for general domains in $\mathbb{R}^d$.
- The graph Laplacian *eigenvalues* reflect various intrinsic geometric and topological information about the graph including connectivity or the number of separated components; diameter; mean distance, . . .
- Fan Chung: *Spectral Graph Theory*, Amer. Math. Soc., 1997, says: *"This monograph is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs."*
- Due to the time limitation, I will *not* be able to discuss the details on how the graph Laplacian *eigenvalues* reveal the geometric and topological information of the graph. For the details, please check the above book, the books listed in the beginning of this section, and
  - R. Merris: "Laplacian matrices of graphs: a survey," *Linear Algebra Appl.*, vol. 197/198, pp. 143–176, 1994.
  - N. Saito & E. Woei: "Analysis of neuronal dendrite patterns using eigenvalues of graph Laplacians," *Japan SIAM Lett.* vol. 1 pp. 13–16, 2009 (Invited paper).

Why Graph Laplacians?

- We already know that the Laplacian eigenvalues and eigenfunctions are extremely useful for general domains in $\mathbb{R}^d$.
- The graph Laplacian *eigenvalues* reflect various intrinsic geometric and topological information about the graph including connectivity or the number of separated components; diameter; mean distance, . . .
- Fan Chung: *Spectral Graph Theory*, Amer. Math. Soc., 1997, says: *"This monograph is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs."*
- Due to the time limitation, I will *not* be able to discuss the details on how the graph Laplacian *eigenvalues* reveal the geometric and topological information of the graph. For the details, please check the above book, the books listed in the beginning of this section, and
  - R. Merris: "Laplacian matrices of graphs: a survey," *Linear Algebra Appl.*, vol. 197/198, pp. 143–176, 1994.
  - N. Saito & E. Woei: "Analysis of neuronal dendrite patterns using eigenvalues of graph Laplacians," *Japan SIAM Lett.* vol. 1 pp. 13–16, 2009 (Invited paper).

Why Graph Laplacians?

- We already know that the Laplacian eigenvalues and eigenfunctions are extremely useful for general domains in $\mathbb{R}^d$.
- The graph Laplacian *eigenvalues* reflect various intrinsic geometric and topological information about the graph including connectivity or the number of separated components; diameter; mean distance, . . .
- Fan Chung: *Spectral Graph Theory*, Amer. Math. Soc., 1997, says: *"This monograph is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs."*
- Due to the time limitation, I will *not* be able to discuss the details on how the graph Laplacian *eigenvalues* reveal the geometric and topological information of the graph. For the details, please check the above book, the books listed in the beginning of this section, and
  - R. Merris: "Laplacian matrices of graphs: a survey," *Linear Algebra Appl.*, vol. 197/198, pp. 143–176, 1994.
  - N. Saito & E. Woei: "Analysis of neuronal dendrite patterns using eigenvalues of graph Laplacians," *Japan SIAM Lett.* vol. 1 pp. 13–16, 2009 (Invited paper).

Why Graph Laplacians?

- We already know that the Laplacian eigenvalues and eigenfunctions are extremely useful for general domains in $\mathbb{R}^d$.
- The graph Laplacian *eigenvalues* reflect various intrinsic geometric and topological information about the graph including connectivity or the number of separated components; diameter; mean distance, . . .
- Fan Chung: *Spectral Graph Theory*, Amer. Math. Soc., 1997, says: *"This monograph is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs."*
- Due to the time limitation, I will *not* be able to discuss the details on how the graph Laplacian *eigenvalues* reveal the geometric and topological information of the graph. For the details, please check the above book, the books listed in the beginning of this section, and
    - R. Merris: "Laplacian matrices of graphs: a survey," *Linear Algebra Appl.*, vol. 197/198, pp. 143–176, 1994.
    - N. Saito & E. Woei: "Analysis of neuronal dendrite patterns using eigenvalues of graph Laplacians," *Japan SIAM Lett.* vol. 1 pp. 13–16, 2009 (Invited paper).

Why Graph Laplacians?

- We already know that the Laplacian eigenvalues and eigenfunctions are extremely useful for general domains in $\mathbb{R}^d$.
- The graph Laplacian *eigenvalues* reflect various intrinsic geometric and topological information about the graph including connectivity or the number of separated components; diameter; mean distance, ...
- Fan Chung: *Spectral Graph Theory*, Amer. Math. Soc., 1997, says: *"This monograph is an intertwined tale of eigenvalues and their use in unlocking a thousand secrets about graphs."*
- Due to the time limitation, I will *not* be able to discuss the details on how the graph Laplacian *eigenvalues* reveal the geometric and topological information of the graph. For the details, please check the above book, the books listed in the beginning of this section, and
    - R. Merris: "Laplacian matrices of graphs: a survey," *Linear Algebra Appl.*, vol. 197/198, pp. 143–176, 1994.
    - N. Saito & E. Woei: "Analysis of neuronal dendrite patterns using eigenvalues of graph Laplacians," *Japan SIAM Lett.* vol. 1 pp. 13–16, 2009 (Invited paper).

Why Graph Laplacians? ...

- The graph Laplacian *eigenfunctions* form an orthonormal basis on a graph $\Longrightarrow$
  - can *expand* functions defined on a graph
  - can perform *spectral analysis/synthesis/filtering* of data measured on vertices of a graph
- Can be used for graph partitioning, graph drawing, data analysis, clustering, ... $\Longrightarrow$ Graph Cut, Spectral Clustering
- Less studied than graph Laplacian eigenvalues
- In this tutorial, I will use the terms "eigenfunctions" and "eigenvectors" interchangeably.
- Also, an eigenvector/function is denoted by ϕ, and its value at vertex $x \in V$ is denoted by $\phi(x)$.

Why Graph Laplacians? ...

- The graph Laplacian *eigenfunctions* form an orthonormal basis on a graph $\Longrightarrow$
  - can *expand* functions defined on a graph
  - can perform *spectral analysis/synthesis/filtering* of data measured on vertices of a graph
- Can be used for graph partitioning, graph drawing, data analysis, clustering, ... $\Longrightarrow$ Graph Cut, Spectral Clustering
- Less studied than graph Laplacian eigenvalues
- In this tutorial, I will use the terms "eigenfunctions" and "eigenvectors" interchangeably.
- Also, an eigenvector/function is denoted by ϕ, and its value at vertex $x \in V$ is denoted by $\phi(x)$.

Why Graph Laplacians? ...

- The graph Laplacian *eigenfunctions* form an orthonormal basis on a graph $\Longrightarrow$
  - can *expand* functions defined on a graph
  - can perform *spectral analysis/synthesis/filtering* of data measured on vertices of a graph
- Can be used for graph partitioning, graph drawing, data analysis, clustering, ... $\Longrightarrow$ Graph Cut, Spectral Clustering
- Less studied than graph Laplacian eigenvalues
- In this tutorial, I will use the terms "eigenfunctions" and "eigenvectors" interchangeably.
- Also, an eigenvector/function is denoted by ϕ, and its value at vertex $x \in V$ is denoted by $\phi(x)$.

Why Graph Laplacians? ...

- The graph Laplacian *eigenfunctions* form an orthonormal basis on a graph $\Longrightarrow$
  - can *expand* functions defined on a graph
  - can perform *spectral analysis/synthesis/filtering* of data measured on vertices of a graph
- Can be used for graph partitioning, graph drawing, data analysis, clustering, ... $\Longrightarrow$ Graph Cut, Spectral Clustering
- Less studied than graph Laplacian eigenvalues
- In this tutorial, I will use the terms "eigenfunctions" and "eigenvectors" interchangeably.
- Also, an eigenvector/function is denoted by ϕ, and its value at vertex $x \in V$ is denoted by $\phi(x)$.

Why Graph Laplacians? . . .

- The graph Laplacian *eigenfunctions* form an orthonormal basis on a graph $\Longrightarrow$
  - can *expand* functions defined on a graph
  - can perform *spectral analysis/synthesis/filtering* of data measured on vertices of a graph
- Can be used for graph partitioning, graph drawing, data analysis, clustering, . . . $\Longrightarrow$ Graph Cut, Spectral Clustering
- Less studied than graph Laplacian eigenvalues
- In this tutorial, I will use the terms "eigenfunctions" and "eigenvectors" interchangeably.
- Also, an eigenvector/function is denoted by ϕ, and its value at vertex $x \in V$ is denoted by $\phi(x)$.

Why Graph Laplacians? . . .

- The graph Laplacian *eigenfunctions* form an orthonormal basis on a graph $\Longrightarrow$
  - can *expand* functions defined on a graph
  - can perform *spectral analysis/synthesis/filtering* of data measured on vertices of a graph
- Can be used for graph partitioning, graph drawing, data analysis, clustering, . . . $\Longrightarrow$ Graph Cut, Spectral Clustering
- Less studied than graph Laplacian eigenvalues
- In this tutorial, I will use the terms "eigenfunctions" and "eigenvectors" interchangeably.
- Also, an eigenvector/function is denoted by ϕ, and its value at vertex $x \in V$ is denoted by $\phi(x)$.

Why Graph Laplacians? ...

- The graph Laplacian *eigenfunctions* form an orthonormal basis on a graph $\Longrightarrow$
  - can *expand* functions defined on a graph
  - can perform *spectral analysis/synthesis/filtering* of data measured on vertices of a graph
- Can be used for graph partitioning, graph drawing, data analysis, clustering, ... $\Longrightarrow$ Graph Cut, Spectral Clustering
- Less studied than graph Laplacian eigenvalues
- In this tutorial, I will use the terms "eigenfunctions" and "eigenvectors" interchangeably.
- Also, an eigenvector/function is denoted by $\boldsymbol{\phi}$, and its value at vertex $x \in V$ is denoted by $\boldsymbol{\phi}(x)$.

A Simple Yet Important Example: A Path Graph

$$\underbrace{\begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}}_{L(G)} = \underbrace{\begin{bmatrix} 1 & & & & & \\ & 2 & & & & \\ & & 2 & & & \\ & & & \ddots & & \\ & & & & 2 & \\ & & & & & 1 \end{bmatrix}}_{D(G)} - \underbrace{\begin{bmatrix} 0 & 1 & & & & \\ 1 & 0 & 1 & & & \\ & 1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 0 & 1 \\ & & & & 1 & 0 \end{bmatrix}}_{A(G)}$$

The eigenvectors of this matrix are exactly the DCT Type II basis vectors used for the JPEG image compression standard! (See G. Strang, "The discrete cosine transform," *SIAM Review*, vol. 41, pp. 135–147, 1999).

- $\lambda_k = 2 - 2\cos(\pi k/n) = 4\sin^2(\pi k/2n)$, $k = 0, 1, \ldots, n-1$.
- $\boldsymbol{\phi}_k(\ell) = \cos\left(\pi k\left(\ell + \frac{1}{2}\right)/n\right)$, $k, \ell = 0, 1, \ldots, n-1$.
- In this simple case, λ (eigenvalue) is a monotonic function w.r.t. the frequency, which is the eigenvalue index k. However, in general, the notion of frequency is not well defined.

Outline

A Brief Review of Graph Laplacian Eigenvalues

- In this review part, we only consider undirected and unweighted graphs and their unnormalized Laplacians $L(G) = D(G) - A(G)$. Let $|V(G)| = n$, $|E(G)| = m$.
- It is a good exercise to see how the statements change for the *normalized* or *symmetrically-normalized* graph Laplacians.
- Can show that $L(G)$ is positive semi-definite.
- Hence, we can *sort* the eigenvalues of $L(G)$ as $0 = \lambda_0(G) \le \lambda_1(G) \le \cdots \le \lambda_{n-1}(G)$ and denote the set of these eigenvalue by $\Lambda(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.

A Brief Review of Graph Laplacian Eigenvalues

- In this review part, we only consider undirected and unweighted graphs and their unnormalized Laplacians $L(G) = D(G) - A(G)$. Let $|V(G)| = n$, $|E(G)| = m$.
- It is a good exercise to see how the statements change for the *normalized* or *symmetrically-normalized* graph Laplacians.
- Can show that $L(G)$ is positive semi-definite.
- Hence, we can *sort* the eigenvalues of $L(G)$ as $0 = \lambda_0(G) \le \lambda_1(G) \le \cdots \le \lambda_{n-1}(G)$ and denote the set of these eigenvalue by $\Lambda(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.

A Brief Review of Graph Laplacian Eigenvalues

- In this review part, we only consider undirected and unweighted graphs and their unnormalized Laplacians $L(G) = D(G) - A(G)$. Let $|V(G)| = n$, $|E(G)| = m$.
- It is a good exercise to see how the statements change for the *normalized* or *symmetrically-normalized* graph Laplacians.
- Can show that $L(G)$ is positive semi-definite.
- Hence, we can *sort* the eigenvalues of $L(G)$ as $0 = \lambda_0(G) \le \lambda_1(G) \le \cdots \le \lambda_{n-1}(G)$ and denote the set of these eigenvalue by $\Lambda(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.

A Brief Review of Graph Laplacian Eigenvalues

- In this review part, we only consider undirected and unweighted graphs and their unnormalized Laplacians $L(G) = D(G) - A(G)$. Let $|V(G)| = n$, $|E(G)| = m$.
- It is a good exercise to see how the statements change for the *normalized* or *symmetrically-normalized* graph Laplacians.
- Can show that $L(G)$ is positive semi-definite.
- Hence, we can *sort* the eigenvalues of $L(G)$ as $0 = \lambda_0(G) \le \lambda_1(G) \le \cdots \le \lambda_{n-1}(G)$ and denote the set of these eigenvalue by $\Lambda(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.

A Brief Review of Graph Laplacian Eigenvalues

- In this review part, we only consider undirected and unweighted graphs and their unnormalized Laplacians $L(G) = D(G) - A(G)$. Let $|V(G)| = n$, $|E(G)| = m$.
- It is a good exercise to see how the statements change for the *normalized* or *symmetrically-normalized* graph Laplacians.
- Can show that $L(G)$ is positive semi-definite.
- Hence, we can *sort* the eigenvalues of $L(G)$ as $0 = \lambda_0(G) \le \lambda_1(G) \le \cdots \le \lambda_{n-1}(G)$ and denote the set of these eigenvalue by $\Lambda(G)$.
- $m_G(\lambda)$:= the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.

A Brief Review of Graph Laplacian Eigenvalues

- In this review part, we only consider undirected and unweighted graphs and their unnormalized Laplacians $L(G) = D(G) - A(G)$. Let $|V(G)| = n$, $|E(G)| = m$.
- It is a good exercise to see how the statements change for the *normalized* or *symmetrically-normalized* graph Laplacians.
- Can show that $L(G)$ is positive semi-definite.
- Hence, we can *sort* the eigenvalues of $L(G)$ as $0 = \lambda_0(G) \le \lambda_1(G) \le \cdots \le \lambda_{n-1}(G)$ and denote the set of these eigenvalue by $\Lambda(G)$.
- $m_G(\lambda) :=$ the multiplicity of λ.
- Let $I \subset \mathbb{R}$ be an interval of the real line. Then define $m_G(I) := \#\{\lambda_k(G) \in I\}$.

A Brief Review of Graph Laplacian Eigenvalues . . .

- Graph Laplacian matrices of the same graph are permutation-similar. In fact, graphs G_1 and G_2 are *isomorphic* iff there exists a permutation matrix Q such that

$$L(G_2) = Q^{\mathsf{T}} L(G_1) Q.$$

- $\operatorname{rank} L(G) = n - m_G(0)$ where $m_G(0)$ turns out to be the number of connected components of G. Easy to check that $L(G)$ becomes $m_G(0)$ diagonal blocks, and the eigenspace corresponding to the zero eigenvalues is spanned by the *indicator* vectors of each connected component.
- In particular, $\lambda_1 \neq 0$, i.e., $m_G(0) = 1$ iff G is connected.
- This led M. Fiedler (1973) to define the algebraic connectivity of G by $a(G) := \lambda_1(G)$, viewing it as *a quantitative measure of connectivity*.

A Brief Review of Graph Laplacian Eigenvalues . . .

- Graph Laplacian matrices of the same graph are permutation-similar. In fact, graphs G_1 and G_2 are *isomorphic* iff there exists a permutation matrix Q such that
$$L(G_2) = Q^{\mathsf{T}} L(G_1) Q.$$
- $\operatorname{rank} L(G) = n - m_G(0)$ where $m_G(0)$ turns out to be the number of connected components of G. Easy to check that $L(G)$ becomes $m_G(0)$ diagonal blocks, and the eigenspace corresponding to the zero eigenvalues is spanned by the *indicator* vectors of each connected component.
- In particular, $\lambda_1 \neq 0$, i.e., $m_G(0) = 1$ iff G is connected.
- This led M. Fiedler (1973) to define the algebraic connectivity of G by $a(G) := \lambda_1(G)$, viewing it as *a quantitative measure of connectivity*.

A Brief Review of Graph Laplacian Eigenvalues . . .

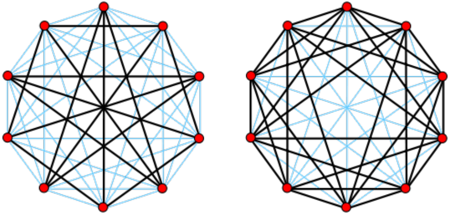
- Graph Laplacian matrices of the same graph are permutation-similar. In fact, graphs G_1 and G_2 are *isomorphic* iff there exists a permutation matrix Q such that
$$L(G_2) = Q^{\mathsf{T}} L(G_1) Q.$$
- $\operatorname{rank} L(G) = n - m_G(0)$ where $m_G(0)$ turns out to be the number of connected components of G. Easy to check that $L(G)$ becomes $m_G(0)$ diagonal blocks, and the eigenspace corresponding to the zero eigenvalues is spanned by the *indicator* vectors of each connected component.
- In particular, $\lambda_1 \neq 0$, i.e., $m_G(0) = 1$ iff G is connected.
- This led M. Fiedler (1973) to define the algebraic connectivity of G by $a(G) := \lambda_1(G)$, viewing it as *a quantitative measure of connectivity*.

A Brief Review of Graph Laplacian Eigenvalues . . .

- Graph Laplacian matrices of the same graph are permutation-similar. In fact, graphs G_1 and G_2 are *isomorphic* iff there exists a permutation matrix Q such that
$$L(G_2) = Q^{\mathsf{T}} L(G_1) Q.$$
- $\operatorname{rank} L(G) = n - m_G(0)$ where $m_G(0)$ turns out to be the number of connected components of G. Easy to check that $L(G)$ becomes $m_G(0)$ diagonal blocks, and the eigenspace corresponding to the zero eigenvalues is spanned by the *indicator* vectors of each connected component.
- In particular, $\lambda_1 \neq 0$, i.e., $m_G(0) = 1$ iff G is connected.
- This led M. Fiedler (1973) to define the algebraic connectivity of G by $a(G) := \lambda_1(G)$, viewing it as *a quantitative measure of connectivity*.

A Brief Review of Graph Laplacian Eigenvalues . . .

- Denote the complement of G (in K_n) by G^c.



The Petersen graph and its complement in K_{10} (from Wikipedia)

- Then, we have

$$L(G) + L(G^c) = L(K_n) = nI_n - J_n,$$

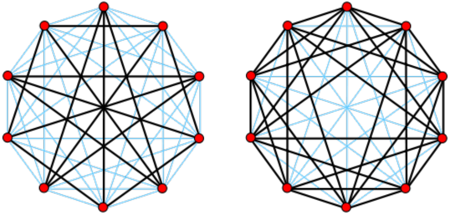
where J_n is the $n \times n$ matrix whose entries are all 1.

- We also have:

$$\Lambda(G^c) = \{0, n - \lambda_{n-1}(G), n - \lambda_{n-2}(G), \ldots, n - \lambda_1(G)\}.$$

A Brief Review of Graph Laplacian Eigenvalues . . .

- Denote the complement of G (in K_n) by G^c.



The Petersen graph and its complement in K_{10} (from Wikipedia)

- Then, we have

$$L(G) + L(G^c) = L(K_n) = nI_n - J_n,$$

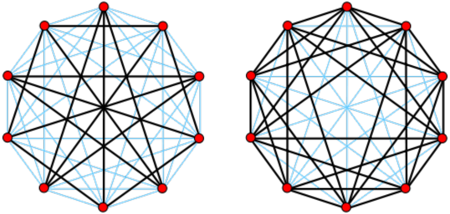
where J_n is the $n \times n$ matrix whose entries are all 1.

- We also have:

$$\Lambda(G^c) = \{0, n - \lambda_{n-1}(G), n - \lambda_{n-2}(G), \ldots, n - \lambda_1(G)\}.$$

A Brief Review of Graph Laplacian Eigenvalues . . .

- Denote the complement of G (in K_n) by G^c.



The Petersen graph and its complement in K_{10} (from Wikipedia)

- Then, we have

$$L(G) + L(G^c) = L(K_n) = nI_n - J_n,$$

where J_n is the $n \times n$ matrix whose entries are all 1.

- We also have:

$$\Lambda(G^c) = \{0, n - \lambda_{n-1}(G), n - \lambda_{n-2}(G), \ldots, n - \lambda_1(G)\}.$$

A Brief Review of Graph Laplacian Eigenvalues . . .

- From the above, we can see that

$$\lambda_{\max}(G) = \lambda_{n-1}(G) \leq n,$$

and $m_G(n) = m_{G^c}(0) - 1$.

- On the other hand, Grone and Merris showed in 1994

$$\lambda_{\max}(G) = \lambda_{n-1}(G) \geq \max_{1 \leq j \leq n} d_j + 1.$$

- Let G be a connected graph and suppose $L(G)$ has exactly k distinct eigenvalues. Then

$$\mathrm{diam}(G) \leq k - 1.$$

A Brief Review of Graph Laplacian Eigenvalues ...

- From the above, we can see that

$$\lambda_{\max}(G) = \lambda_{n-1}(G) \le n,$$

and $m_G(n) = m_{G^c}(0) - 1$.

- On the other hand, Grone and Merris showed in 1994

$$\lambda_{\max}(G) = \lambda_{n-1}(G) \ge \max_{1 \le j \le n} d_j + 1.$$

- Let G be a connected graph and suppose $L(G)$ has exactly k distinct eigenvalues. Then

$$\operatorname{diam}(G) \le k - 1.$$

A Brief Review of Graph Laplacian Eigenvalues . . .

- From the above, we can see that

$$\lambda_{\max}(G) = \lambda_{n-1}(G) \le n,$$

  and $m_G(n) = m_{G^c}(0) - 1$.

- On the other hand, Grone and Merris showed in 1994

$$\lambda_{\max}(G) = \lambda_{n-1}(G) \ge \max_{1 \le j \le n} d_j + 1.$$

- Let G be a connected graph and suppose $L(G)$ has exactly k distinct eigenvalues. Then

$$\mathrm{diam}(G) \le k - 1.$$

Outline

Basic Properties of GL Eigenfunctions

- If $G = (V, E)$, $|V| = n$, is connected, then $\lambda_0 = 0$, $a(G) = \lambda_1 > 0$.
- We already know that the eigenfunction corresponding to $\lambda_0 = 0$ is $\boldsymbol{\phi}_0 = \mathbf{1}_n$.
- Hence, $\boldsymbol{\phi}_j$ corresponding to $\lambda_j > 0$, $j = 1, \dots, n-1$, must be orthogonal to $\mathbf{1}_n$: $\sum_{x \in V} \boldsymbol{\phi}_j(x) = 0$, i.e., it must *oscillate*.
- If $\boldsymbol{\phi}(x) = 0$, then $(L\boldsymbol{\phi})(x) = \lambda \boldsymbol{\phi}(x) = 0$. Hence, $\sum_{y \sim x} L_{xy} \boldsymbol{\phi}(y) = 0$.

Theorem (Grover (1990); Gladwell & Zhu (2002))

An eigenfunction of $L(G)$ cannot have a nonnegative local minimum or a nonpositive local maximum.

Proof. Suppose $\boldsymbol{\phi}(x)$ is a local minimum of $\boldsymbol{\phi}$ with $\boldsymbol{\phi}(x) \geq 0$. Then, $\forall y \sim x$, $\boldsymbol{\phi}(x) - \boldsymbol{\phi}(y) < 0$. Now, recall $L\boldsymbol{\phi}(x) = \sum_{y \sim x} a_{xy}(\boldsymbol{\phi}(x) - \boldsymbol{\phi}(y)) = \lambda \boldsymbol{\phi}(x) \geq 0$ where $a_{xy} \geq 0$ is the xy-th entry of the adjacency matrix $A(G)$. These contradicts each other. □

Basic Properties of GL Eigenfunctions

- If $G=(V,E)$, $|V|=n$, is connected, then $\lambda_0=0$, $a(G)=\lambda_1>0$.
- We already know that the eigenfunction corresponding to $\lambda_0=0$ is $\boldsymbol{\phi}_0=\mathbf{1}_n$.
- Hence, $\boldsymbol{\phi}_j$ corresponding to $\lambda_j>0$, $j=1,\ldots,n-1$, must be orthogonal to $\mathbf{1}_n$: $\sum_{x\in V}\boldsymbol{\phi}_j(x)=0$, i.e., it must *oscillate*.
- If $\boldsymbol{\phi}(x)=0$, then $(L\boldsymbol{\phi})(x)=\lambda\boldsymbol{\phi}(x)=0$. Hence, $\sum_{y\sim x}L_{xy}\boldsymbol{\phi}(y)=0$.

Theorem (Grover (1990); Gladwell & Zhu (2002))

An eigenfunction of $L(G)$ cannot have a nonnegative local minimum or a nonpositive local maximum.

Proof. Suppose $\boldsymbol{\phi}(x)$ is a local minimum of $\boldsymbol{\phi}$ with $\boldsymbol{\phi}(x)\geq 0$. Then, $\forall y\sim x$, $\boldsymbol{\phi}(x)-\boldsymbol{\phi}(y)<0$. Now, recall $L\boldsymbol{\phi}(x)=\sum_{y\sim x}a_{xy}(\boldsymbol{\phi}(x)-\boldsymbol{\phi}(y))=\lambda\boldsymbol{\phi}(x)\geq 0$ where $a_{xy}\geq 0$ is the xy-th entry of the adjacency matrix $A(G)$. These contradicts each other. □

Basic Properties of GL Eigenfunctions

- If $G = (V, E)$, $|V| = n$, is connected, then $\lambda_0 = 0$, $a(G) = \lambda_1 > 0$.
- We already know that the eigenfunction corresponding to $\lambda_0 = 0$ is $\boldsymbol{\phi}_0 = \mathbf{1}_n$.
- Hence, $\boldsymbol{\phi}_j$ corresponding to $\lambda_j > 0$, $j = 1, \ldots, n-1$, must be orthogonal to $\mathbf{1}_n$: $\sum_{x \in V} \boldsymbol{\phi}_j(x) = 0$, i.e., it must *oscillate*.
- If $\phi(x) = 0$, then $(L\phi)(x) = \lambda\phi(x) = 0$. Hence, $\sum_{y \sim x} L_{xy}\phi(y) = 0$.

Theorem (Grover (1990); Gladwell & Zhu (2002))

An eigenfunction of $L(G)$ cannot have a nonnegative local minimum or a nonpositive local maximum.

Proof. Suppose $\phi(x)$ is a local minimum of ϕ with $\phi(x) \geq 0$. Then, $\forall y \sim x$, $\phi(x) - \phi(y) < 0$. Now, recall $L\phi(x) = \sum_{y \sim x} a_{xy}(\phi(x) - \phi(y)) = \lambda\phi(x) \geq 0$ where $a_{xy} \geq 0$ is the xy-th entry of the adjacency matrix $A(G)$. These contradicts each other. □

Basic Properties of GL Eigenfunctions

- If $G = (V, E)$, $|V| = n$, is connected, then $\lambda_0 = 0$, $a(G) = \lambda_1 > 0$.
- We already know that the eigenfunction corresponding to $\lambda_0 = 0$ is $\boldsymbol{\phi}_0 = \mathbf{1}_n$.
- Hence, $\boldsymbol{\phi}_j$ corresponding to $\lambda_j > 0$, $j = 1, \ldots, n-1$, must be orthogonal to $\mathbf{1}_n$: $\sum_{x \in V} \boldsymbol{\phi}_j(x) = 0$, i.e., it must *oscillate*.
- If $\boldsymbol{\phi}(x) = 0$, then $(L\boldsymbol{\phi})(x) = \lambda\boldsymbol{\phi}(x) = 0$. Hence, $\sum_{y \sim x} L_{xy}\boldsymbol{\phi}(y) = 0$.

Theorem (Grover (1990); Gladwell & Zhu (2002))

An eigenfunction of $L(G)$ cannot have a nonnegative local minimum or a nonpositive local maximum.

Proof. Suppose $\boldsymbol{\phi}(x)$ is a local minimum of $\boldsymbol{\phi}$ with $\boldsymbol{\phi}(x) \geq 0$. Then, $\forall y \sim x$, $\boldsymbol{\phi}(x) - \boldsymbol{\phi}(y) < 0$. Now, recall $L\boldsymbol{\phi}(x) = \sum_{y \sim x} a_{xy}(\boldsymbol{\phi}(x) - \boldsymbol{\phi}(y)) = \lambda\boldsymbol{\phi}(x) \geq 0$ where $a_{xy} \geq 0$ is the xy-th entry of the adjacency matrix $A(G)$. These contradicts each other. □

Basic Properties of GL Eigenfunctions

- If $G = (V, E)$, $|V| = n$, is connected, then $\lambda_0 = 0$, $a(G) = \lambda_1 > 0$.
- We already know that the eigenfunction corresponding to $\lambda_0 = 0$ is $\boldsymbol{\phi}_0 = \mathbf{1}_n$.
- Hence, $\boldsymbol{\phi}_j$ corresponding to $\lambda_j > 0$, $j = 1, \ldots, n-1$, must be orthogonal to $\mathbf{1}_n$: $\sum_{x \in V} \boldsymbol{\phi}_j(x) = 0$, i.e., it must *oscillate*.
- If $\boldsymbol{\phi}(x) = 0$, then $(L\boldsymbol{\phi})(x) = \lambda\boldsymbol{\phi}(x) = 0$. Hence, $\sum_{y \sim x} L_{xy}\boldsymbol{\phi}(y) = 0$.

Theorem (Grover (1990); Gladwell & Zhu (2002))

An eigenfunction of $L(G)$ cannot have a nonnegative local minimum or a nonpositive local maximum.

Proof. Suppose $\boldsymbol{\phi}(x)$ is a local minimum of $\boldsymbol{\phi}$ with $\boldsymbol{\phi}(x) \geq 0$. Then, $\forall y \sim x$, $\boldsymbol{\phi}(x) - \boldsymbol{\phi}(y) < 0$. Now, recall $L\boldsymbol{\phi}(x) = \sum_{y \sim x} a_{xy}(\boldsymbol{\phi}(x) - \boldsymbol{\phi}(y)) = \lambda\boldsymbol{\phi}(x) \geq 0$ where $a_{xy} \geq 0$ is the xy-th entry of the adjacency matrix $A(G)$. These contradicts each other. □

Basic Properties of GL Eigenfunctions

- If $G = (V, E)$, $|V| = n$, is connected, then $\lambda_0 = 0$, $a(G) = \lambda_1 > 0$.
- We already know that the eigenfunction corresponding to $\lambda_0 = 0$ is $\boldsymbol{\phi}_0 = \mathbf{1}_n$.
- Hence, $\boldsymbol{\phi}_j$ corresponding to $\lambda_j > 0$, $j = 1, \ldots, n-1$, must be orthogonal to $\mathbf{1}_n$: $\sum_{x \in V} \boldsymbol{\phi}_j(x) = 0$, i.e., it must *oscillate*.
- If $\boldsymbol{\phi}(x) = 0$, then $(L\boldsymbol{\phi})(x) = \lambda\boldsymbol{\phi}(x) = 0$. Hence, $\sum_{y \sim x} L_{xy}\boldsymbol{\phi}(y) = 0$.

Theorem (Grover (1990); Gladwell & Zhu (2002))

An eigenfunction of $L(G)$ cannot have a nonnegative local minimum or a nonpositive local maximum.

Proof. Suppose $\boldsymbol{\phi}(x)$ is a local minimum of $\boldsymbol{\phi}$ with $\boldsymbol{\phi}(x) \geq 0$. Then, $\forall y \sim x$, $\boldsymbol{\phi}(x) - \boldsymbol{\phi}(y) < 0$. Now, recall $L\boldsymbol{\phi}(x) = \sum_{y \sim x} a_{xy}(\boldsymbol{\phi}(x) - \boldsymbol{\phi}(y)) = \lambda\boldsymbol{\phi}(x) \geq 0$ where $a_{xy} \geq 0$ is the xy-th entry of the adjacency matrix $A(G)$. These contradicts each other. □

Basic Properties of Unweighted GL Eigenfunctions

Theorem (Merris (1998))

If $0 \lneqq \lambda < n$ is an eigenvalue of $L(G)$, then any eigenfunction affording λ takes the value 0 on every vertex of degree $n-1$.

Proof. Let $v \in V$ be a vertex with $d(v) = n-1$. Then, $L\phi(v) = (n-1)\phi(v) - \sum_{u \neq v} \phi(u) = \lambda\phi(v)$. But, $\phi \perp \mathbf{1}_n$, so $\sum_{u \neq v} \phi(u) = -\phi(v)$. This leads to: $n\phi(v) = \lambda\phi(v)$. Since $0 \lneqq \lambda \lneqq n$, we must have $\phi(v) = 0$. □

Theorem (Merris (1998))

Let (λ, ϕ) be an eigenpair of $L(G)$. If $\phi(u) = \phi(v)$, then (λ, ϕ) is also an eigenpair of $L(G')$ where G' is the graph obtained from G by either deleting or adding the edge $e = (u, v)$ depending on whether or not $e \in E(G)$.

Basic Properties of Unweighted GL Eigenfunctions

Theorem (Merris (1998))

If $0 \lneqq \lambda < n$ is an eigenvalue of $L(G)$, then any eigenfunction affording λ takes the value 0 on every vertex of degree $n-1$.

Proof. Let $v \in V$ be a vertex with $d(v) = n-1$. Then, $L\boldsymbol{\phi}(v) = (n-1)\boldsymbol{\phi}(v) - \sum_{u \neq v} \boldsymbol{\phi}(u) = \lambda\boldsymbol{\phi}(v)$. But, $\boldsymbol{\phi} \perp \mathbf{1}_n$, so $\sum_{u \neq v} \boldsymbol{\phi}(u) = -\boldsymbol{\phi}(v)$. This leads to: $n\boldsymbol{\phi}(v) = \lambda\boldsymbol{\phi}(v)$. Since $0 \lneqq \lambda \lneqq n$, we must have $\boldsymbol{\phi}(v) = 0$. □

Theorem (Merris (1998))

Let $(\lambda, \boldsymbol{\phi})$ be an eigenpair of $L(G)$. If $\boldsymbol{\phi}(u) = \boldsymbol{\phi}(v)$, then $(\lambda, \boldsymbol{\phi})$ is also an eigenpair of $L(G')$ where G' is the graph obtained from G by either deleting or adding the edge $e = (u, v)$ depending on whether or not $e \in E(G)$.

Basic Properties of Unweighted GL Eigenfunctions

Theorem (Merris (1998))

If $0 \lneqq \lambda < n$ is an eigenvalue of $L(G)$, then any eigenfunction affording λ takes the value 0 on every vertex of degree $n-1$.

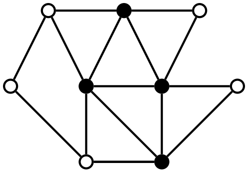
Proof. Let $v \in V$ be a vertex with $d(v) = n-1$. Then, $L\boldsymbol{\phi}(v) = (n-1)\boldsymbol{\phi}(v) - \sum_{u \neq v} \boldsymbol{\phi}(u) = \lambda\boldsymbol{\phi}(v)$. But, $\boldsymbol{\phi} \perp \mathbf{1}_n$, so $\sum_{u \neq v} \boldsymbol{\phi}(u) = -\boldsymbol{\phi}(v)$. This leads to: $n\boldsymbol{\phi}(v) = \lambda\boldsymbol{\phi}(v)$. Since $0 \lneqq \lambda \lneqq n$, we must have $\boldsymbol{\phi}(v) = 0$. □

Theorem (Merris (1998))

Let $(\lambda, \boldsymbol{\phi})$ be an eigenpair of $L(G)$. If $\boldsymbol{\phi}(u) = \boldsymbol{\phi}(v)$, then $(\lambda, \boldsymbol{\phi})$ is also an eigenpair of $L(G')$ where G' is the graph obtained from G by either deleting or adding the edge $e = (u, v)$ depending on whether or not $e \in E(G)$.

Basic Properties of Unweighted GL Eigenfunctions . . .

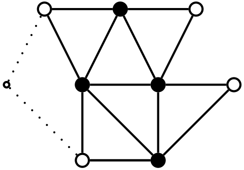
Let W be a nonempty subset of $V(G)$. Then, the reduced graph $G\{W\}$ is obtained from G by deleting all vertices in $V \setminus W$ that are not adjacent to a vertex of W and subsequent deletion of any remaining edges that are not incident with a vertex in W.



$W = \{\bullet\},\ W^c = \{\circ\}$

Basic Properties of Unweighted GL Eigenfunctions . . .

Let W be a nonempty subset of $V(G)$. Then, the reduced graph $G\{W\}$ is obtained from G by deleting all vertices in $V \setminus W$ that are not adjacent to a vertex of W and subsequent deletion of any remaining edges that are not incident with a vertex in W.



$G\{W\}$

Basic Properties of Unweighted GL Eigenfunctions ...

Theorem (Merris (1998))

Fix a nonempty subset $W \subset V$. Suppose $\boldsymbol{\phi}$ is an eigenfunction of the reduced graph $G\{W\}$ that affords λ and is supported by W in the sense that if $\boldsymbol{\phi}(u) \neq 0$, then $u \in W$. Then the extension $\widetilde{\boldsymbol{\phi}}$ with $\widetilde{\boldsymbol{\phi}}(v) = \boldsymbol{\phi}(v)$ for $v \in W$ and $\widetilde{\boldsymbol{\phi}}(v) = 0$ for $v \in V \setminus W$ is an eigenfunction of G affording λ.

Theorem (Merris (1998))

Let $\boldsymbol{\phi}$ be an eigenfunction affording λ of G. Let N_v be the set of neighbors of v. Suppose $\boldsymbol{\phi}(u) = \boldsymbol{\phi}(v) = 0$, where $N_u \cap N_v = \emptyset$. Let G' be the graph on $n-1$ vertices obtained by coalescing u and v into a single vertex, which is adjacent in G' to precisely those vertices that are adjacent in G to u or to v. Then, the function $\boldsymbol{\phi}'$ obtained by restricting $\boldsymbol{\phi}$ to $V(G) \setminus \{v\}$ is an eigenfunction of G' affording λ.

Basic Properties of Unweighted GL Eigenfunctions ...

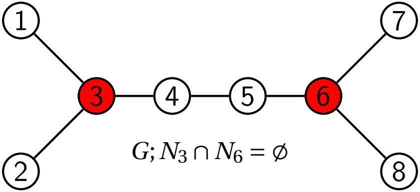
Theorem (Merris (1998))

Fix a nonempty subset $W \subset V$. *Suppose* $\boldsymbol{\phi}$ *is an eigenfunction of the reduced graph* $G\{W\}$ *that affords* λ *and is supported by* W *in the sense that if* $\boldsymbol{\phi}(u) \neq 0$, *then* $u \in W$. *Then the extension* $\widetilde{\boldsymbol{\phi}}$ *with* $\widetilde{\boldsymbol{\phi}}(v) = \boldsymbol{\phi}(v)$ *for* $v \in W$ *and* $\widetilde{\boldsymbol{\phi}}(v) = 0$ *for* $v \in V \setminus W$ *is an eigenfunction of* G *affording* λ.

Theorem (Merris (1998))

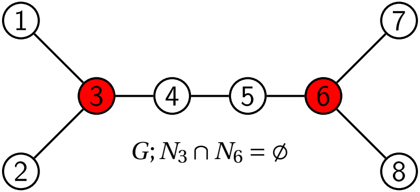
Let $\boldsymbol{\phi}$ *be an eigenfunction affording* λ *of* G. *Let* N_v *be the set of neighbors of* v. *Suppose* $\boldsymbol{\phi}(u) = \boldsymbol{\phi}(v) = 0$, *where* $N_u \cap N_v = \emptyset$. *Let* G' *be the graph on* $n-1$ *vertices obtained by coalescing* u *and* v *into a single vertex, which is adjacent in* G' *to precisely those vertices that are adjacent in* G *to* u *or to* v. *Then, the function* $\boldsymbol{\phi}'$ *obtained by restricting* $\boldsymbol{\phi}$ *to* $V(G) \setminus \{v\}$ *is an eigenfunction of* G' *affording* λ.

A Simple Example

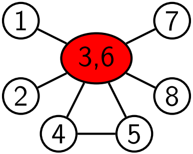


$$\lambda_2(G) = 1; \boldsymbol{\phi}_2(G) = [-0.0261, -0.0261, {\color{red}0}, 0.0523, 0.0523, {\color{red}0}, -0.7303, 0.6781]^{\mathsf{T}}$$

A Simple Example



$$\lambda_2(G) = 1; \boldsymbol{\phi}_2(G) = [-0.0261, -0.0261, 0, 0.0523, 0.0523, 0, -0.7303, 0.6781]^{\mathsf{T}}$$

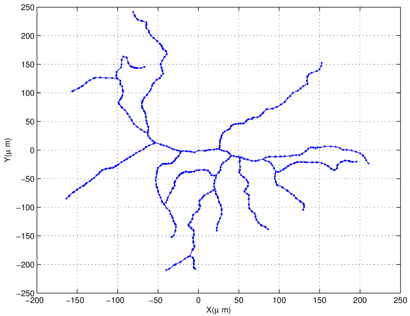


$$\lambda_2(G') = 1; \boldsymbol{\phi}_2(G') \propto [-0.0261, -0.0261, 0, 0.0523, 0.0523, -0.7303, 0.6781]^{\mathsf{T}}$$

Outline

A Peculiar Phase Transition Phenomenon

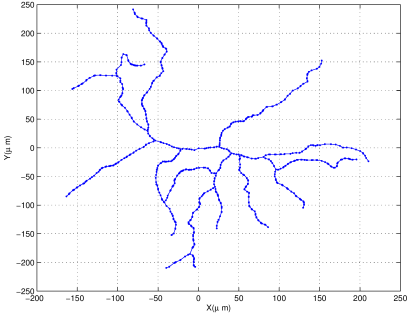
We observed an interesting phase transition phenomenon on the behavior of the eigenvalues of *graph Laplacians* defined on dendritic trees.



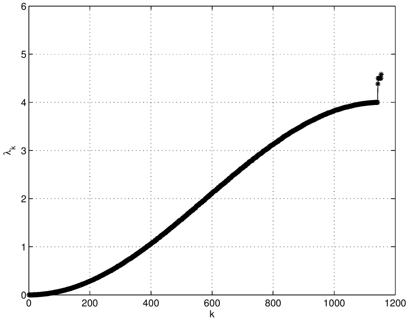
(a) RGC #100

A Peculiar Phase Transition Phenomenon

We observed an interesting phase transition phenomenon on the behavior of the eigenvalues of *graph Laplacians* defined on dendritic trees.



(a) RGC #100



(b) Eigenvalues of RGC #100

A Peculiar Phase Transition Phenomenon . . .

We have observed that this value 4 is critical since:

- the eigenfunctions corresponding to the eigenvalues below 4 are *semi-global oscillations* (like *Fourier cosines/sines*) over the entire dendrites or one of the dendrite arbors;
- those corresponding to the eigenvalues above 4 are much more *localized* (like *wavelets*) around *junctions/bifurcation vertices*.

A Peculiar Phase Transition Phenomenon ...

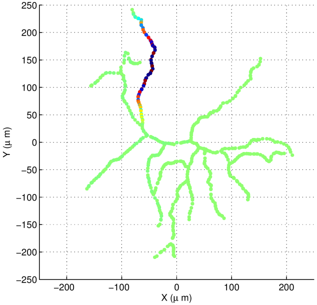
We have observed that this value 4 is critical since:

- the eigenfunctions corresponding to the eigenvalues below 4 are *semi-global oscillations* (like *Fourier cosines/sines*) over the entire dendrites or one of the dendrite arbors;
- those corresponding to the eigenvalues above 4 are much more *localized* (like *wavelets*) around *junctions/bifurcation vertices*.

A Peculiar Phase Transition Phenomenon . . .

We have observed that this value 4 is critical since:

- the eigenfunctions corresponding to the eigenvalues below 4 are *semi-global oscillations* (like *Fourier cosines/sines*) over the entire dendrites or one of the dendrite arbors;
- those corresponding to the eigenvalues above 4 are much more *localized* (like *wavelets*) around *junctions/bifurcation vertices.*

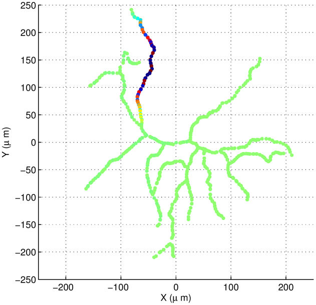


(a) RGC #100; $\lambda_{1141} = 3.9994$

A Peculiar Phase Transition Phenomenon . . .

We have observed that this value 4 is critical since:

- the eigenfunctions corresponding to the eigenvalues below 4 are *semi-global oscillations* (like *Fourier cosines/sines*) over the entire dendrites or one of the dendrite arbors;
- those corresponding to the eigenvalues above 4 are much more *localized* (like *wavelets*) around *junctions/bifurcation vertices*.

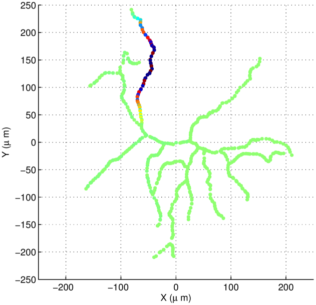


(a) RGC #100; $\lambda_{1141} = 3.9994$

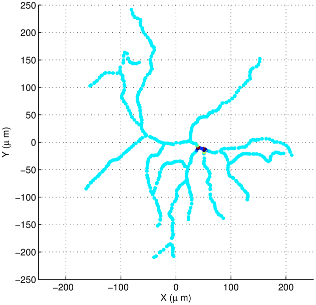
A Peculiar Phase Transition Phenomenon . . .

We have observed that this value 4 is critical since:

- the eigenfunctions corresponding to the eigenvalues below 4 are *semi-global oscillations* (like *Fourier cosines/sines*) over the entire dendrites or one of the dendrite arbors;
- those corresponding to the eigenvalues above 4 are much more *localized* (like *wavelets*) around *junctions/bifurcation vertices*.



(a) RGC #100; $\lambda_{1141} = 3.9994$



(b) RGC #100; $\lambda_{1142} = 4.3829$

- We know why such localization/phase transition occurs $\Longrightarrow$ See our article for the detail: Y. Nakatsukasa, N. Saito, & E. Woei: "Mysteries around graph Laplacian eigenvalue 4," *Linear Algebra & Its Applications*, vol. 438, no. 8, pp. 3231–3246, 2013.
- Any physiological consequence? Importance of branching vertices?
- Many such eigenvector localization phenomena have been reported: Anderson localization, scars in quantum chaos, ...
- See also an interesting related work for more general setting and for application in numerical linear algebra: I. Krishtal, T. Strohmer, & T. Wertz: "Localization of matrix factorizations," *Foundations of Comp. Math.*, to appear, 2014.
- Our point is that *eigenvectors corresponding to high eigenvalues are quite sensitive to topology and geometry of the underlying domain and cannot really be viewed as high frequency oscillations unless the underlying graph is a simple unweighted path.*

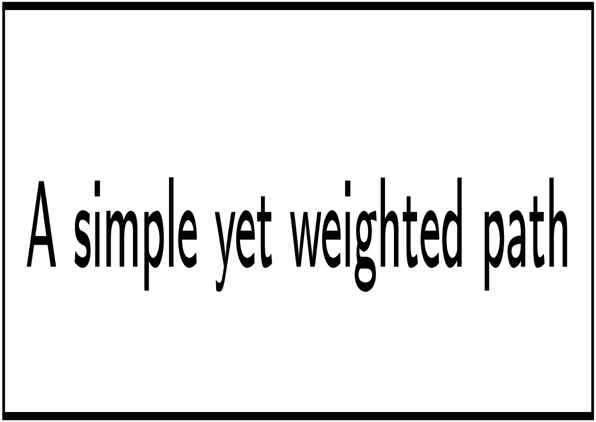
- We know why such localization/phase transition occurs $\Longrightarrow$ See our article for the detail: Y. Nakatsukasa, N. Saito, & E. Woei: "Mysteries around graph Laplacian eigenvalue 4," *Linear Algebra & Its Applications*, vol. 438, no. 8, pp. 3231–3246, 2013.
- Any physiological consequence? Importance of branching vertices?
- Many such eigenvector localization phenomena have been reported: Anderson localization, scars in quantum chaos, ...
- See also an interesting related work for more general setting and for application in numerical linear algebra: I. Krishtal, T. Strohmer, & T. Wertz: "Localization of matrix factorizations," *Foundations of Comp. Math.*, to appear, 2014.
- Our point is that *eigenvectors corresponding to high eigenvalues are quite sensitive to topology and geometry of the underlying domain and cannot really be viewed as high frequency oscillations unless the underlying graph is a simple unweighted path.*

- We know why such localization/phase transition occurs $\Longrightarrow$ See our article for the detail: Y. Nakatsukasa, N. Saito, & E. Woei: "Mysteries around graph Laplacian eigenvalue 4," *Linear Algebra & Its Applications*, vol. 438, no. 8, pp. 3231–3246, 2013.
- Any physiological consequence? Importance of branching vertices?
- Many such eigenvector localization phenomena have been reported: Anderson localization, scars in quantum chaos, ...
- See also an interesting related work for more general setting and for application in numerical linear algebra: I. Krishtal, T. Strohmer, & T. Wertz: "Localization of matrix factorizations," *Foundations of Comp. Math.*, to appear, 2014.
- Our point is that *eigenvectors corresponding to high eigenvalues are quite sensitive to topology and geometry of the underlying domain and cannot really be viewed as high frequency oscillations unless the underlying graph is a simple unweighted path.*

- We know why such localization/phase transition occurs $\Longrightarrow$ See our article for the detail: Y. Nakatsukasa, N. Saito, & E. Woei: "Mysteries around graph Laplacian eigenvalue 4," *Linear Algebra & Its Applications*, vol. 438, no. 8, pp. 3231–3246, 2013.
- Any physiological consequence? Importance of branching vertices?
- Many such eigenvector localization phenomena have been reported: Anderson localization, scars in quantum chaos, ...
- See also an interesting related work for more general setting and for application in numerical linear algebra: I. Krishtal, T. Strohmer, & T. Wertz: "Localization of matrix factorizations," *Foundations of Comp. Math.*, to appear, 2014.
- Our point is that *eigenvectors corresponding to high eigenvalues are quite sensitive to topology and geometry of the underlying domain and cannot really be viewed as high frequency oscillations unless the underlying graph is a simple unweighted path.*

- We know why such localization/phase transition occurs $\Longrightarrow$ See our article for the detail: Y. Nakatsukasa, N. Saito, & E. Woei: "Mysteries around graph Laplacian eigenvalue 4," *Linear Algebra & Its Applications*, vol. 438, no. 8, pp. 3231–3246, 2013.
- Any physiological consequence? Importance of branching vertices?
- Many such eigenvector localization phenomena have been reported: Anderson localization, scars in quantum chaos, . . .
- See also an interesting related work for more general setting and for application in numerical linear algebra: I. Krishtal, T. Strohmer, & T. Wertz: "Localization of matrix factorizations," *Foundations of Comp. Math.*, to appear, 2014.
- Our point is that *eigenvectors corresponding to high eigenvalues are quite sensitive to topology and geometry of the underlying domain and cannot really be viewed as high frequency oscillations unless the underlying graph is a simple unweighted path.*

- Even a simple path, if edges are weighted, localization tends to occur.



Play/Pause

- We want to control such eigenvector localizations by ourselves rather than dictated by the topology and geometry of the graphs!
- This leads us to the development of the *multiscale basis dictionaries* on graphs.

Outline

Graph Partitioning via Spectral Clustering

Goal: Split the vertices V into two "good" subsets, X and X^c

Plan: Use the signs of the entries in $\boldsymbol{\phi}_1$ known as the Fiedler vector

Why? Using $\boldsymbol{\phi}_1$ to generate X and X^c yields an approximate minimizer of the RatioCut function[1,2]:

$$\mathrm{RatioCut}(X, X^c) := \frac{\mathrm{cut}(X, X^c)}{|X|} + \frac{\mathrm{cut}(X, X^c)}{|X^c|},$$

where

$$\mathrm{cut}(X, X^c) := \sum_{\substack{v_i \in X \\ v_j \in X^c}} A_{ij}$$

[1] L. Hagen and A. B. Kahng: "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des.*, vol. 11, no. 9, pp. 1074-1085, 1992.

[2] We could also use the signs of $\boldsymbol{\phi}_1$ of $L_{\mathrm{rw}} := D^{-1}L$ (equivalently, $L_{\mathrm{sym}} := D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$), which yield an approximate minimizer of the *Normalized Cut* function of Shi and Malik.

Graph Partitioning via Spectral Clustering

Goal: Split the vertices V into two "good" subsets, X and X^c

Plan: Use the signs of the entries in $\boldsymbol{\phi}_1$ known as the Fiedler vector

Why? Using $\boldsymbol{\phi}_1$ to generate X and X^c yields an approximate minimizer of the RatioCut function[1,2]:

$$\mathrm{RatioCut}(X, X^c) := \frac{\mathrm{cut}(X, X^c)}{|X|} + \frac{\mathrm{cut}(X, X^c)}{|X^c|},$$

where

$$\mathrm{cut}(X, X^c) := \sum_{\substack{v_i \in X \\ v_j \in X^c}} A_{ij}$$

[1] L. Hagen and A. B. Kahng: "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des.*, vol. 11, no. 9, pp. 1074-1085, 1992.

[2] We could also use the signs of $\boldsymbol{\phi}_1$ of $L_{\mathrm{rw}} := D^{-1}L$ (equivalently, $L_{\mathrm{sym}} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$), which yield an approximate minimizer of the *Normalized Cut* function of Shi and Malik.

Graph Partitioning via Spectral Clustering

Goal: Split the vertices V into two "good" subsets, X and X^c

Plan: Use the signs of the entries in $\boldsymbol{\phi}_1$ known as the Fiedler vector

Why? Using $\boldsymbol{\phi}_1$ to generate X and X^c yields an approximate minimizer of the RatioCut function[1,2]:

$$\mathrm{RatioCut}(X, X^c) := \frac{\mathrm{cut}(X, X^c)}{|X|} + \frac{\mathrm{cut}(X, X^c)}{|X^c|},$$

where

$$\mathrm{cut}(X, X^c) := \sum_{\substack{v_i \in X \\ v_j \in X^c}} A_{ij}$$

[1] L. Hagen and A. B. Kahng: "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des.*, vol. 11, no. 9, pp. 1074-1085, 1992.

[2] We could also use the signs of $\boldsymbol{\phi}_1$ of $L_{\mathrm{rw}} := D^{-1}L$ (equivalently, $L_{\mathrm{sym}} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$), which yield an approximate minimizer of the *Normalized Cut* function of Shi and Malik.

Example of Graph Partitioning

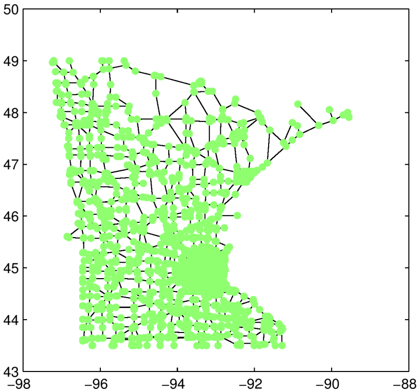


Figure: The MN road network

Example of Graph Partitioning

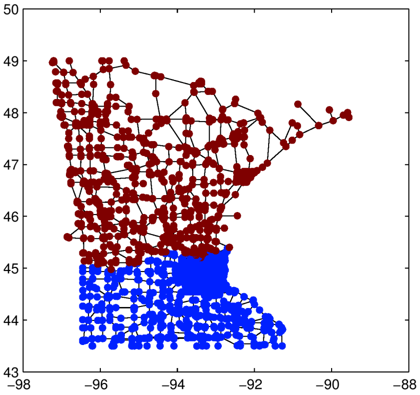


Figure: The MN road network partitioned via the Fiedler vector of L

Outline

Motivation: Building Multiscale Basis Dictionaries

- *Wavelets* have been quite successful on regular domains
- They have been extended to irregular domains ⇒ "2nd Generation Wavelets" including graphs, e.g.:
  - Coifman and Maggioni (2006): diffusion wavelets; Bremer *et al.* (2006): diffusion wavelet *packets*
  - Jansen, Nason, and Silverman (2008): Adaptation of the *lifting scheme* to graphs
  - Hammond, Vandergheynst, and Gribonval (2011): Spectral graph wavelet transforms (via spectral graph theory)
  - ...
- **Key difficulties:**
  - The notion of *frequency* is ill-defined on graphs and the Fourier transform is not properly defined on graphs
  - Hence, the use of graph Laplacian eigenvectors, which can be viewed as "sines" and "cosines" on graphs, has been quite popular
  - However, they exhibit peculiar behaviors depending on *topology* and *structure* of given graphs!

Motivation: Building Multiscale Basis Dictionaries

- *Wavelets* have been quite successful on regular domains
- They have been extended to irregular domains ⇒ "2nd Generation Wavelets" including graphs, e.g.:
  - Coifman and Maggioni (2006): diffusion wavelets; Bremer *et al.* (2006): diffusion wavelet *packets*
  - Jansen, Nason, and Silverman (2008): Adaptation of the *lifting scheme* to graphs
  - Hammond, Vandergheynst, and Gribonval (2011): Spectral graph wavelet transforms (via spectral graph theory)
  - . . .
- Key difficulties:
  - The notion of *frequency* is ill-defined on graphs and the Fourier transform is not properly defined on graphs
  - Hence, the use of graph Laplacian eigenvectors, which can be viewed as "sines" and "cosines" on graphs, has been quite popular
  - However, they exhibit peculiar behaviors depending on *topology* and *structure* of given graphs!

Motivation: Building Multiscale Basis Dictionaries

- *Wavelets* have been quite successful on regular domains
- They have been extended to irregular domains ⇒ "2nd Generation Wavelets" including graphs, e.g.:
  - Coifman and Maggioni (2006): diffusion wavelets; Bremer *et al.* (2006): diffusion wavelet *packets*
  - Jansen, Nason, and Silverman (2008): Adaptation of the *lifting scheme* to graphs
  - Hammond, Vandergheynst, and Gribonval (2011): Spectral graph wavelet transforms (via spectral graph theory)
  - ...
- **Key difficulties:**
  - The notion of *frequency* is ill-defined on graphs and the Fourier transform is not properly defined on graphs
  - Hence, the use of graph Laplacian eigenvectors, which can be viewed as "sines" and "cosines" on graphs, has been quite popular
  - However, they exhibit peculiar behaviors depending on *topology* and *structure* of given graphs!

Our transforms involve 2 main steps:

1. Recursively partition the graph

⇕ These steps can be performed concurrently, or we can fully partition the graph and then generate a set of bases

2. Using the regions on each level of the graph partitioning, generate a set of orthonormal bases for the graph

Our transforms involve 2 main steps:

1. Recursively partition the graph

⇕ These steps can be performed concurrently, or we can fully partition the graph and then generate a set of bases

2. Using the regions on each level of the graph partitioning, generate a set of orthonormal bases for the graph

Outline

Hierarchical Graph Laplacian Eigen Transform (HGLET)

Now we present a novel transform that can be viewed as a generalization of the *block Discrete Cosine Transform*. We refer to this transform as the *Hierarchical Graph Laplacian Eigen Transform (HGLET)*.

The algorithm proceeds as follows...

1. Generate an orthonormal basis for the entire graph ⇒ Laplacian eigenvectors (Notation is $\boldsymbol{\phi}^{j}_{k,l}$ with $j = 0$)
2. Partition the graph using the Fiedler vector $\boldsymbol{\phi}^{j}_{k,1}$
3. Generate an orthonormal basis for each of the partitions ⇒ Laplacian eigenvectors
4. Repeat...

$$\left[\begin{array}{ccccc} \boldsymbol{\phi}^{0}_{0,0} & \boldsymbol{\phi}^{0}_{0,1} & \boldsymbol{\phi}^{0}_{0,2} & \cdots & \boldsymbol{\phi}^{0}_{0,N^0_0-1} \end{array}\right]$$

1. Generate an orthonormal basis for the entire graph ⇒ Laplacian eigenvectors (Notation is $\boldsymbol{\phi}^{j}_{k,l}$ with $j = 0$)
2. Partition the graph using the Fiedler vector $\boldsymbol{\phi}^{j}_{k,1}$
3. Generate an orthonormal basis for each of the partitions ⇒ Laplacian eigenvectors
4. Repeat...

$$\left[\begin{array}{ccccc} \boldsymbol{\phi}^{0}_{0,0} & \boldsymbol{\phi}^{0}_{0,1} & \boldsymbol{\phi}^{0}_{0,2} & \cdots & \boldsymbol{\phi}^{0}_{0,N^0_0-1} \end{array}\right]$$

1. Generate an orthonormal basis for the entire graph ⇒ Laplacian eigenvectors (Notation is $\boldsymbol{\phi}^{j}_{k,l}$ with $j = 0$)
2. Partition the graph using the Fiedler vector $\boldsymbol{\phi}^{j}_{k,1}$
3. Generate an orthonormal basis for each of the partitions ⇒ Laplacian eigenvectors
4. Repeat...

$$\left[\ \boldsymbol{\phi}^{0}_{0,0} \quad \boldsymbol{\phi}^{0}_{0,1} \quad \boldsymbol{\phi}^{0}_{0,2} \quad \cdots \quad \boldsymbol{\phi}^{0}_{0,N^0_0-1}\ \right]$$

$$\left[\ \boldsymbol{\phi}^{1}_{0,0} \ \boldsymbol{\phi}^{1}_{0,1} \ \boldsymbol{\phi}^{1}_{0,2} \ \cdots \ \boldsymbol{\phi}^{1}_{0,N^1_0-1}\ \right] \quad \left[\ \boldsymbol{\phi}^{1}_{1,0} \ \boldsymbol{\phi}^{1}_{1,1} \ \boldsymbol{\phi}^{1}_{1,2} \ \cdots \ \boldsymbol{\phi}^{1}_{1,N^1_1-1}\ \right]$$

1. Generate an orthonormal basis for the entire graph ⇒ Laplacian eigenvectors (Notation is $\boldsymbol{\phi}^{j}_{k,l}$ with $j = 0$)
2. Partition the graph using the Fiedler vector $\boldsymbol{\phi}^{j}_{k,1}$
3. Generate an orthonormal basis for each of the partitions ⇒ Laplacian eigenvectors
4. Repeat...

$$\begin{bmatrix} \boldsymbol{\phi}^{0}_{0,0} & \boldsymbol{\phi}^{0}_{0,1} & \boldsymbol{\phi}^{0}_{0,2} & \cdots & \boldsymbol{\phi}^{0}_{0,N^0_0-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^{1}_{0,0} & \boldsymbol{\phi}^{1}_{0,1} & \boldsymbol{\phi}^{1}_{0,2} & \cdots & \boldsymbol{\phi}^{1}_{0,N^1_0-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^{1}_{1,0} & \boldsymbol{\phi}^{1}_{1,1} & \boldsymbol{\phi}^{1}_{1,2} & \cdots & \boldsymbol{\phi}^{1}_{1,N^1_1-1} \end{bmatrix}$$

1. Generate an orthonormal basis for the entire graph ⇒ Laplacian eigenvectors (Notation is $\boldsymbol{\phi}^{j}_{k,l}$ with $j = 0$)
2. Partition the graph using the Fiedler vector $\boldsymbol{\phi}^{j}_{k,1}$
3. Generate an orthonormal basis for each of the partitions ⇒ Laplacian eigenvectors
4. Repeat...

$$\left[\begin{array}{ccccc} \boldsymbol{\phi}^{0}_{0,0} & \boldsymbol{\phi}^{0}_{0,1} & \boldsymbol{\phi}^{0}_{0,2} & \cdots & \boldsymbol{\phi}^{0}_{0,N^0_0-1} \end{array}\right]$$

$$\left[\begin{array}{ccccc} \boldsymbol{\phi}^{1}_{0,0} & \boldsymbol{\phi}^{1}_{0,1} & \boldsymbol{\phi}^{1}_{0,2} & \cdots & \boldsymbol{\phi}^{1}_{0,N^1_0-1} \end{array}\right] \quad \left[\begin{array}{ccccc} \boldsymbol{\phi}^{1}_{1,0} & \boldsymbol{\phi}^{1}_{1,1} & \boldsymbol{\phi}^{1}_{1,2} & \cdots & \boldsymbol{\phi}^{1}_{1,N^1_1-1} \end{array}\right]$$

1. Generate an orthonormal basis for the entire graph ⇒ Laplacian eigenvectors (Notation is $\boldsymbol{\phi}^j_{k,l}$ with $j = 0$)
2. Partition the graph using the Fiedler vector $\boldsymbol{\phi}^j_{k,1}$
3. Generate an orthonormal basis for each of the partitions ⇒ Laplacian eigenvectors
4. Repeat...

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{0,0}\, \boldsymbol{\phi}^2_{0,1} \cdots \boldsymbol{\phi}^2_{0,N^2_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{1,0}\, \boldsymbol{\phi}^2_{1,1} \cdots \boldsymbol{\phi}^2_{1,N^2_1-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{2,0}\, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{3,0}\, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$

1. Generate an orthonormal basis for the entire graph ⇒ Laplacian eigenvectors (Notation is $\boldsymbol{\phi}^{j}_{k,l}$ with $j = 0$)
2. Partition the graph using the Fiedler vector $\boldsymbol{\phi}^{j}_{k,1}$
3. Generate an orthonormal basis for each of the partitions ⇒ Laplacian eigenvectors
4. Repeat...

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{0,0}\, \boldsymbol{\phi}^2_{0,1} \cdots \boldsymbol{\phi}^2_{0,N^2_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{1,0}\, \boldsymbol{\phi}^2_{1,1} \cdots \boldsymbol{\phi}^2_{1,N^2_1-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{2,0}\, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{3,0}\, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$

1. Generate an orthonormal basis for the entire graph ⇒ Laplacian eigenvectors (Notation is $\boldsymbol{\phi}^{j}_{k,l}$ with $j = 0$)
2. Partition the graph using the Fiedler vector $\boldsymbol{\phi}^{j}_{k,1}$
3. Generate an orthonormal basis for each of the partitions ⇒ Laplacian eigenvectors
4. Repeat...

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{0,0}\, \boldsymbol{\phi}^2_{0,1} \cdots \boldsymbol{\phi}^2_{0,N^2_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{1,0}\, \boldsymbol{\phi}^2_{1,1} \cdots \boldsymbol{\phi}^2_{1,N^2_1-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{2,0}\, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{3,0}\, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$

$$\vdots$$

Remarks

- For an unweighted path graph, this exactly yields a dictionary of the block DCT-II
- Similar to wavelet packet or local cosine dictionaries in that it generates an *overcomplete basis* from which we can select a basis useful for the task at hand ⇒ best-basis algorithm, local discriminant basis algorithm, . . .
  - A union of bases on disjoint subsets is obviously orthonormal

Remarks

- For an unweighted path graph, this exactly yields a dictionary of the block DCT-II
- Similar to wavelet packet or local cosine dictionaries in that it generates an *overcomplete basis* from which we can select a basis useful for the task at hand ⇒ best-basis algorithm, local discriminant basis algorithm, . . .
    - A union of bases on disjoint subsets is obviously orthonormal

Remarks

- For an unweighted path graph, this exactly yields a dictionary of the block DCT-II
- Similar to wavelet packet or local cosine dictionaries in that it generates an *overcomplete basis* from which we can select a basis useful for the task at hand ⇒ best-basis algorithm, local discriminant basis algorithm, . . .
    - A union of bases on disjoint subsets is obviously orthonormal

Remarks

- For an unweighted path graph, this exactly yields a dictionary of the block DCT-II
- Similar to wavelet packet or local cosine dictionaries in that it generates an *overcomplete basis* from which we can select a basis useful for the task at hand ⇒ best-basis algorithm, local discriminant basis algorithm, . . .
  - A union of bases on disjoint subsets is obviously orthonormal

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{0,0} & \cdots & \boldsymbol{\phi}^2_{0,N^2_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{1,0} & \cdots & \boldsymbol{\phi}^2_{1,N^2_1-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{2,0} & \cdots & \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} & \cdots & \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$

Remarks

- For an unweighted path graph, this exactly yields a dictionary of the block DCT-II
- Similar to wavelet packet or local cosine dictionaries in that it generates an *overcomplete basis* from which we can select a basis useful for the task at hand ⇒ best-basis algorithm, local discriminant basis algorithm, . . .
    - A union of bases on disjoint subsets is obviously orthonormal

$$\left[\boldsymbol{\phi}^0_{0,0} \quad \boldsymbol{\phi}^0_{0,1} \quad \boldsymbol{\phi}^0_{0,2} \quad \cdots \quad \boldsymbol{\phi}^0_{0,N^0_0-1} \right]$$

$$\left[\boldsymbol{\phi}^1_{0,0} \quad \boldsymbol{\phi}^1_{0,1} \quad \boldsymbol{\phi}^1_{0,2} \quad \cdots \quad \boldsymbol{\phi}^1_{0,N^1_0-1} \right] \left[\boldsymbol{\phi}^1_{1,0} \quad \boldsymbol{\phi}^1_{1,1} \quad \boldsymbol{\phi}^1_{1,2} \quad \cdots \quad \boldsymbol{\phi}^1_{1,N^1_1-1} \right]$$

$$\left[\boldsymbol{\phi}^2_{0,0} \quad \cdots \quad \boldsymbol{\phi}^2_{0,N^2_0-1} \right] \left[\boldsymbol{\phi}^2_{1,0} \quad \cdots \quad \boldsymbol{\phi}^2_{1,N^2_1-1} \right] \left[\boldsymbol{\phi}^2_{2,0} \quad \cdots \quad \boldsymbol{\phi}^2_{2,N^2_2-1} \right] \left[\boldsymbol{\phi}^2_{3,0} \quad \cdots \quad \boldsymbol{\phi}^2_{3,N^2_3-1} \right]$$

Remarks

- For an unweighted path graph, this exactly yields a dictionary of the block DCT-II
- Similar to wavelet packet or local cosine dictionaries in that it generates an *overcomplete basis* from which we can select a basis useful for the task at hand ⇒ best-basis algorithm, local discriminant basis algorithm, ...
  - A union of bases on disjoint subsets is obviously orthonormal

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{0,0} & \cdots & \boldsymbol{\phi}^2_{0,N^2_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{1,0} & \cdots & \boldsymbol{\phi}^2_{1,N^2_1-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{2,0} & \cdots & \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} & \cdots & \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$

Remarks

- For an unweighted path graph, this exactly yields a dictionary of the block DCT-II
- Similar to wavelet packet or local cosine dictionaries in that it generates an *overcomplete basis* from which we can select a basis useful for the task at hand ⇒ best-basis algorithm, local discriminant basis algorithm, . . .
    - A union of bases on disjoint subsets is obviously orthonormal

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{0,0} & \cdots & \boldsymbol{\phi}^2_{0,N^2_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{1,0} & \cdots & \boldsymbol{\phi}^2_{1,N^2_1-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{2,0} & \cdots & \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} & \cdots & \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$

Remarks

- For an unweighted path graph, this exactly yields a dictionary of the block DCT-II
- Similar to wavelet packet or local cosine dictionaries in that it generates an *overcomplete basis* from which we can select a basis useful for the task at hand ⇒ best-basis algorithm, local discriminant basis algorithm, . . .
    - A union of bases on disjoint subsets is obviously orthonormal

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$

$$\color{red}{\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix}} \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{0,0} & \cdots & \boldsymbol{\phi}^2_{0,N^2_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{1,0} & \cdots & \boldsymbol{\phi}^2_{1,N^2_1-1} \end{bmatrix} \color{red}{\begin{bmatrix} \boldsymbol{\phi}^2_{2,0} & \cdots & \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} & \cdots & \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}}$$

Remarks

- For an unweighted path graph, this exactly yields a dictionary of the block DCT-II
- Similar to wavelet packet or local cosine dictionaries in that it generates an *overcomplete basis* from which we can select a basis useful for the task at hand ⇒ best-basis algorithm, local discriminant basis algorithm, . . .
    - A union of bases on disjoint subsets is obviously orthonormal

$$\left[\; \boldsymbol{\phi}^0_{0,0} \quad \boldsymbol{\phi}^0_{0,1} \quad \boldsymbol{\phi}^0_{0,2} \quad \cdots \quad \boldsymbol{\phi}^0_{0,N^0_0-1} \;\right]$$

$$\left[\boldsymbol{\phi}^1_{0,0} \quad \boldsymbol{\phi}^1_{0,1} \quad \boldsymbol{\phi}^1_{0,2} \quad \cdots \quad \boldsymbol{\phi}^1_{0,N^1_0-1}\right] \left[\boldsymbol{\phi}^1_{1,0} \quad \boldsymbol{\phi}^1_{1,1} \quad \boldsymbol{\phi}^1_{1,2} \quad \cdots \quad \boldsymbol{\phi}^1_{1,N^1_1-1}\right]$$

$$\left[\boldsymbol{\phi}^2_{0,0} \quad \cdots \quad \boldsymbol{\phi}^2_{0,N^2_0-1}\right] \left[\boldsymbol{\phi}^2_{1,0} \quad \cdots \quad \boldsymbol{\phi}^2_{1,N^2_1-1}\right] \left[\boldsymbol{\phi}^2_{2,0} \quad \cdots \quad \boldsymbol{\phi}^2_{2,N^2_2-1}\right] \left[\boldsymbol{\phi}^2_{3,0} \quad \cdots \quad \boldsymbol{\phi}^2_{3,N^2_3-1}\right]$$

Outline

Generalized Haar-Walsh Transform (GHWT)

HGLET is a generalization of the block DCT, and it generates basis vectors that are *smooth on their support*.

The Generalized Haar-Walsh Transform (GHWT) is a generalization of the classical Haar and Walsh-Hadamard Transforms, and it generates basis vectors that are *piecewise-constant on their support*.

The algorithm proceeds as follows...

Generalized Haar-Walsh Transform (GHWT)

HGLET is a generalization of the block DCT, and it generates basis vectors that are *smooth on their support*.

The Generalized Haar-Walsh Transform (GHWT) is a generalization of the classical Haar and Walsh-Hadamard Transforms, and it generates basis vectors that are *piecewise-constant on their support*.

The algorithm proceeds as follows...

1. Generate a full recursive partitioning of the graph ⇒ Fiedler vectors
2. Generate an orthonormal basis for level $j_{\max}$ (the finest level) ⇒ ***scaling vectors*** on the single-node regions
   - As with HGLET, the notation is $\boldsymbol{\psi}^{j}_{k,l}$
3. Using the basis for level $j_{\max}$, generate an orthonormal basis for level $j_{\max}-1$ ⇒ ***scaling*** and ***Haar-like*** vectors
4. Repeat... Using the basis for level j, generate an orthonormal basis for level $j-1$ ⇒ ***scaling***, ***Haar-like***, and ***Walsh-like*** vectors

1. Generate a full recursive partitioning of the graph ⇒ Fiedler vectors
2. Generate an orthonormal basis for level $j_{\max}$ (the finest level) ⇒ ***scaling vectors*** on the single-node regions
   - As with HGLET, the notation is $\boldsymbol{\psi}_{k,l}^{j}$
3. Using the basis for level $j_{\max}$, generate an orthonormal basis for level $j_{\max}-1$ ⇒ ***scaling*** and ***Haar-like*** vectors
4. Repeat... Using the basis for level j, generate an orthonormal basis for level $j-1$ ⇒ ***scaling***, ***Haar-like***, and ***Walsh-like*** vectors

$$\left[\ \boldsymbol{\psi}_{0,0}^{j_{\max}}\ \right] \quad \left[\ \boldsymbol{\psi}_{1,0}^{j_{\max}}\ \right] \quad \left[\ \boldsymbol{\psi}_{2,0}^{j_{\max}}\ \right] \quad \left[\ \boldsymbol{\psi}_{3,0}^{j_{\max}}\ \right] \quad \cdots \quad \left[\ \boldsymbol{\psi}_{K^{j_{\max}}-2,0}^{j_{\max}}\ \right] \qquad \left[\ \boldsymbol{\psi}_{K^{j_{\max}}-1,0}^{j_{\max}}\ \right]$$

1. Generate a full recursive partitioning of the graph $\Rightarrow$ Fiedler vectors
2. Generate an orthonormal basis for level $j_{\max}$ (the finest level) $\Rightarrow$ ***scaling vectors*** on the single-node regions
   - As with HGLET, the notation is $\boldsymbol{\psi}^j_{k,l}$
3. Using the basis for level $j_{\max}$, generate an orthonormal basis for level $j_{\max}-1 \Rightarrow$ ***scaling*** and ***Haar-like*** vectors
4. Repeat... Using the basis for level j, generate an orthonormal basis for level $j-1 \Rightarrow$ ***scaling***, ***Haar-like***, and ***Walsh-like*** vectors

$$\left[\ \boldsymbol{\psi}^{j_{\max}-1}_{0,0}\ \ \boldsymbol{\psi}^{j_{\max}-1}_{0,1}\ \right]\ \left[\ \boldsymbol{\psi}^{j_{\max}-1}_{1,0}\ \ \boldsymbol{\psi}^{j_{\max}-1}_{1,1}\ \right]\ \cdots\ \left[\ \boldsymbol{\psi}^{j_{\max}-1}_{K^{j_{\max}-1}-1,0}\ \ \boldsymbol{\psi}^{j_{\max}-1}_{K^{j_{\max}-1}-1,1}\ \right]$$

$$\left[\ \boldsymbol{\psi}^{j_{\max}}_{0,0}\ \right]\ \left[\ \boldsymbol{\psi}^{j_{\max}}_{1,0}\ \right]\ \left[\ \boldsymbol{\psi}^{j_{\max}}_{2,0}\ \right]\ \left[\ \boldsymbol{\psi}^{j_{\max}}_{3,0}\ \right]\ \cdots\ \left[\ \boldsymbol{\psi}^{j_{\max}}_{K^{j_{\max}}-2,0}\ \right]\ \left[\ \boldsymbol{\psi}^{j_{\max}}_{K^{j_{\max}}-1,0}\ \right]$$

1. Generate a full recursive partitioning of the graph ⇒ Fiedler vectors
2. Generate an orthonormal basis for level $j_{\max}$ (the finest level) ⇒ ***scaling vectors*** on the single-node regions
   - As with HGLET, the notation is $\boldsymbol{\psi}^j_{k,l}$
3. Using the basis for level $j_{\max}$, generate an orthonormal basis for level $j_{\max}-1$ ⇒ ***scaling*** and ***Haar-like*** vectors
4. Repeat... Using the basis for level j, generate an orthonormal basis for level $j-1$ ⇒ ***scaling***, ***Haar-like***, and ***Walsh-like*** vectors

$$\left[\ \boldsymbol{\psi}^0_{0,0} \quad \boldsymbol{\psi}^0_{0,1} \quad \boldsymbol{\psi}^0_{0,2} \quad \boldsymbol{\psi}^0_{0,3} \quad \cdots \quad \boldsymbol{\psi}^0_{0,N-2} \quad \boldsymbol{\psi}^0_{0,N-1}\ \right]$$

$$\vdots$$

$$\left[\ \boldsymbol{\psi}^{j_{\max}-1}_{0,0} \quad \boldsymbol{\psi}^{j_{\max}-1}_{0,1}\ \right] \left[\ \boldsymbol{\psi}^{j_{\max}-1}_{1,0} \quad \boldsymbol{\psi}^{j_{\max}-1}_{1,1}\ \right] \cdots \left[\ \boldsymbol{\psi}^{j_{\max}-1}_{K^{j_{\max}-1}-1,0} \quad \boldsymbol{\psi}^{j_{\max}-1}_{K^{j_{\max}-1}-1,1}\ \right]$$

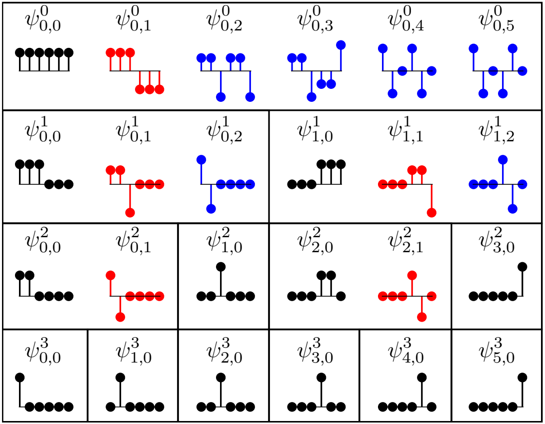
$$\left[\ \boldsymbol{\psi}^{j_{\max}}_{0,0}\ \right] \left[\ \boldsymbol{\psi}^{j_{\max}}_{1,0}\ \right] \left[\ \boldsymbol{\psi}^{j_{\max}}_{2,0}\ \right] \left[\ \boldsymbol{\psi}^{j_{\max}}_{3,0}\ \right] \cdots \left[\ \boldsymbol{\psi}^{j_{\max}}_{K^{j_{\max}}-2,0}\ \right] \left[\ \boldsymbol{\psi}^{j_{\max}}_{K^{j_{\max}}-1,0}\ \right]$$

Remarks

- For an unweighted path graph, this yields a dictionary of Haar-Walsh functions
- As with the HGLET, we can select an orthonormal basis for the entire graph by taking the union of orthonormal bases on disjoint regions

Remarks

- For an unweighted path graph, this yields a dictionary of Haar-Walsh functions
- As with the HGLET, we can select an orthonormal basis for the entire graph by taking the union of orthonormal bases on disjoint regions



Remarks

- We can also reorder and regroup the vectors on each level of the GHWT dictionary according to their type (**scaling**, **Haar-like**, or **Walsh-like**)

- This reorganization gives us *more options for choosing a good basis*

Remarks

- We can also reorder and regroup the vectors on each level of the GHWT dictionary according to their type (**scaling**, **Haar-like**, or **Walsh-like**)

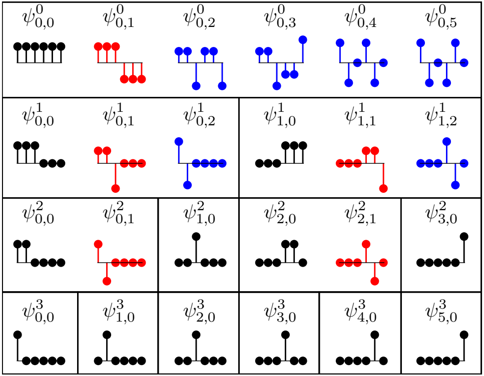


Figure: Default dictionary; i.e., coarse-to-fine

- This reorganization gives us *more options for choosing a good basis*

Remarks

- We can also reorder and regroup the vectors on each level of the GHWT dictionary according to their type (**scaling**, **Haar-like**, or **Walsh-like**)

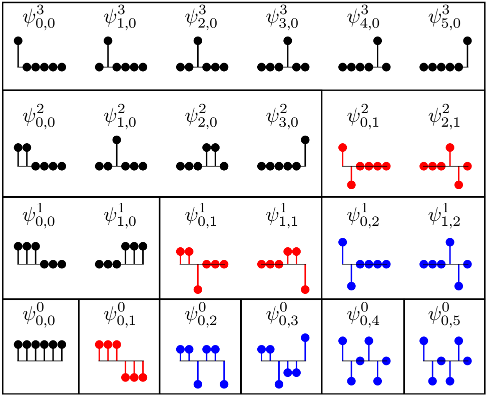


Figure: Reordered & regrouped dictionary; i.e., fine-to-coarse

- This reorganization gives us *more options for choosing a good basis*

Remarks

- We can also reorder and regroup the vectors on each level of the GHWT dictionary according to their type (**scaling**, **Haar-like**, or **Walsh-like**)

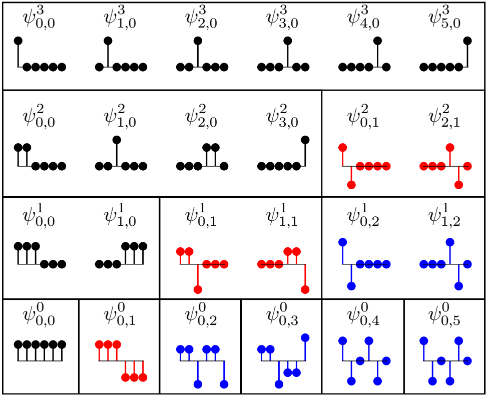


Figure: Reordered & regrouped dictionary; i.e., fine-to-coarse

- This reorganization gives us *more options for choosing a good basis*

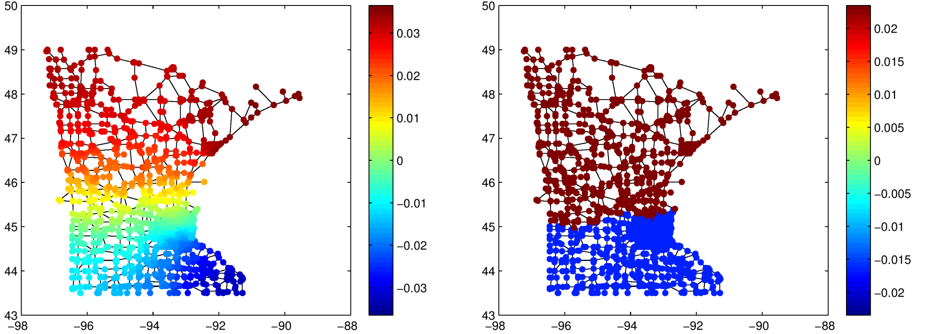
HGLET vs. GHWT

Here we display some of the basis vectors generated by our HGLET (left) and GHWT (right) schemes on the MN road network. (Note: $j = 0$ is the coarsest scale, $j = 14$ is the finest.)

HGLET vs. GHWT

Here we display some of the basis vectors generated by our HGLET (left) and GHWT (right) schemes on the MN road network. (Note: $j = 0$ is the coarsest scale, $j = 14$ is the finest.)

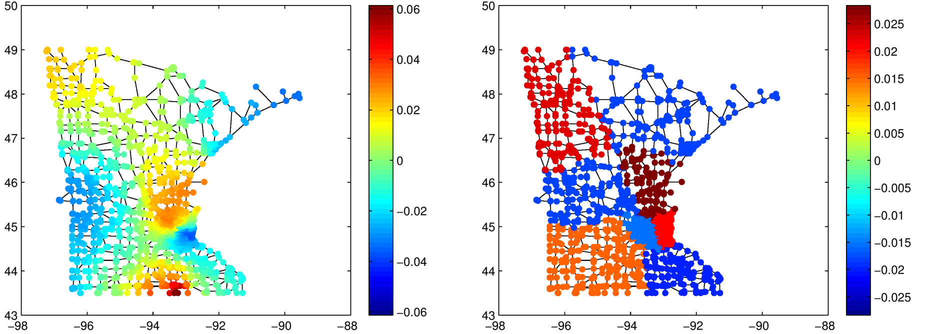
Level $j = 0$, Region $k = 0$, $l = 1$



HGLET vs. GHWT

Here we display some of the basis vectors generated by our HGLET (left) and GHWT (right) schemes on the MN road network. (Note: $j = 0$ is the coarsest scale, $j = 14$ is the finest.)

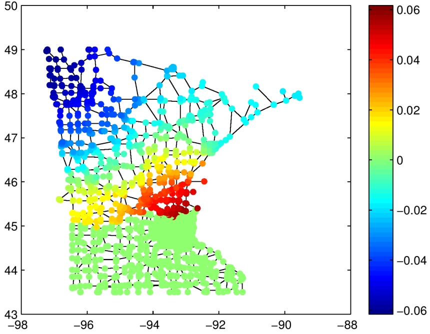
Level $j = 0$, Region $k = 0$, $l = 7$

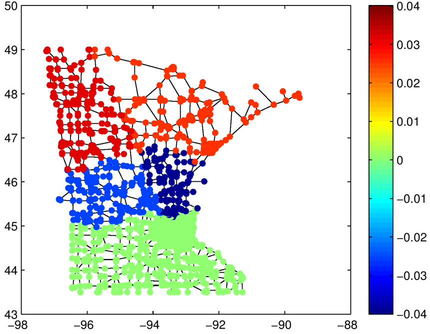


HGLET vs. GHWT

Here we display some of the basis vectors generated by our HGLET (left) and GHWT (right) schemes on the MN road network. (Note: $j = 0$ is the coarsest scale, $j = 14$ is the finest.)

Level $j = 1$, Region $k = 0$, $l = 2$

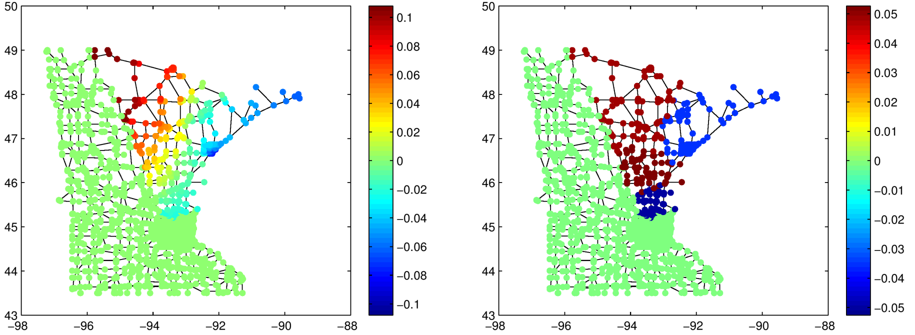




HGLET vs. GHWT

Here we display some of the basis vectors generated by our HGLET (left) and GHWT (right) schemes on the MN road network. (Note: $j = 0$ is the coarsest scale, $j = 14$ is the finest.)

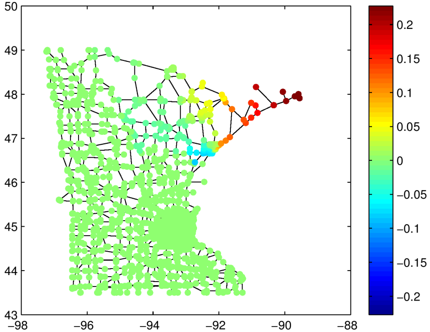
Level $j = 2$, Region $k = 1$, $l = 2$

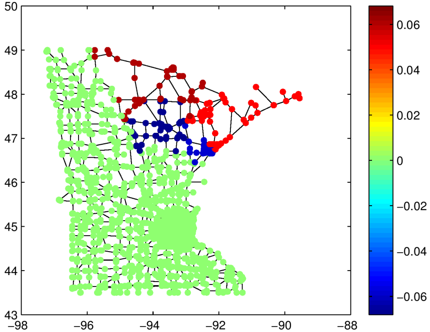


HGLET vs. GHWT

Here we display some of the basis vectors generated by our HGLET (left) and GHWT (right) schemes on the MN road network. (Note: $j = 0$ is the coarsest scale, $j = 14$ is the finest.)

Level $j = 3$, Region $k = 2$, $l = 2$





Computational Complexity: HGLET vs. GHWT

	Computational Complexity	Run Time for MN[1]
HGLET (redundant)	$O(N^3)$	67 sec
GHWT (redundant)	$O(N^2)$	10 sec

[1] Computations performed on a personal laptop (4.00 GB RAM, 2.26 GHz), $N = 2640$ and `nnz(W)` $= 6604$.

Related Work

The following articles also discussed the Haar-like transform on graphs and trees, but *not the Walsh-Hadamard transform* on them:

1. A. D. Szlam, M. Maggioni, R. R. Coifman, and J. C. Bremer, Jr., "Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions," in *Wavelets XI* (M. Papadakis et al. eds.), *Proc. SPIE 5914*, Paper # 59141D, 2005.
2. F. Murtagh, "The Haar wavelet transform of a dendrogram," *J. Classification*, vol. 24, pp. 3–32, 2007.
3. A. Lee, B. Nadler, and L. Wasserman, "Treelets–an adaptive multi-scale basis for sparse unordered data," *Ann. Appl. Stat.*, vol. 2, pp. 435–471, 2008.
4. M. Gavish, B. Nadler, and R. Coifman, "Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning," in *Proc. 27th Intern. Conf. Machine Learning* (J. Fürnkranz et al. eds.), pp. 367–374, Omnipress, Haifa, 2010.

Outline

Coifman and Wickerhauser (1992) developed the best-basis algorithm as a means of selecting the basis from a dictionary of wavelet packets that is "best" for approximation/compression.

We generalize this approach, developing and implementing an algorithm for selecting the basis from the dictionary of HGLET / GHWT bases that is "best" for approximation.

As before, we require a cost functional $\mathscr{J}$. For example:

$$\mathscr{J}(\boldsymbol{x}) = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} = \texttt{norm(x,p)} \quad 0 < p \le 1$$

- For our denoising experiments in the following pages, we used $p = 0.1$.

Coifman and Wickerhauser (1992) developed the best-basis algorithm as a means of selecting the basis from a dictionary of wavelet packets that is "best" for approximation/compression.

We generalize this approach, developing and implementing an algorithm for selecting the basis from the dictionary of HGLET / GHWT bases that is "best" for approximation.

As before, we require a cost functional $\mathscr{J}$. For example:

$$\mathscr{J}(\boldsymbol{x}) = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p} = \texttt{norm(x,p)} \quad 0 < p \le 1$$

- For our denoising experiments in the following pages, we used $p = 0.1$.

Coifman and Wickerhauser (1992) developed the best-basis algorithm as a means of selecting the basis from a dictionary of wavelet packets that is "best" for approximation/compression.

We generalize this approach, developing and implementing an algorithm for selecting the basis from the dictionary of HGLET / GHWT bases that is "best" for approximation.

As before, we require a cost functional $\mathscr{J}$. For example:

$$\mathscr{J}(\boldsymbol{x}) = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} = \texttt{norm(x,p)} \quad 0 < p \leq 1$$

- For our denoising experiments in the following pages, we used $p = 0.1$.

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$
$$d^0_{0,0} \quad d^0_{0,1} \quad d^0_{0,2} \quad \cdots \quad d^0_{0,N^0_0-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$
$$d^1_{0,0} \quad d^1_{0,1} \quad d^1_{0,2} \quad \cdots \quad d^1_{0,N^1_0-1} \qquad d^1_{1,0} \quad d^1_{1,1} \quad d^1_{1,2} \quad \cdots \quad d^1_{1,N^1_1-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{0,0}\, \boldsymbol{\phi}^2_{0,1} \cdots \boldsymbol{\phi}^2_{0,N^2_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{1,0}\, \boldsymbol{\phi}^2_{1,1} \cdots \boldsymbol{\phi}^2_{1,N^2_1-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{2,0}\, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{3,0}\, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$
$$d^2_{0,0}\ d^2_{0,1} \cdots d^2_{0,N^2_0-1} \quad d^2_{1,0}\ d^2_{1,1} \cdots d^2_{1,N^2_1-1} \quad d^2_{2,0}\ d^2_{2,1} \cdots d^2_{2,N^2_2-1} \quad d^2_{3,0}\ d^2_{3,1} \cdots d^2_{3,N^2_3-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$
$$d^0_{0,0} \quad d^0_{0,1} \quad d^0_{0,2} \quad \cdots \quad d^0_{0,N^0_0-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$
$$d^1_{0,0} \quad d^1_{0,1} \quad d^1_{0,2} \quad \cdots \quad d^1_{0,N^1_0-1} \qquad d^1_{1,0} \quad d^1_{1,1} \quad d^1_{1,2} \quad \cdots \quad d^1_{1,N^1_1-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{0,0} \, \boldsymbol{\phi}^2_{0,1} \cdots \boldsymbol{\phi}^2_{0,N^2_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{1,0} \, \boldsymbol{\phi}^2_{1,1} \cdots \boldsymbol{\phi}^2_{1,N^2_1-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{2,0} \, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} \, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$
$$d^2_{0,0} \; d^2_{0,1} \cdots d^2_{0,N^2_0-1} \quad d^2_{1,0} \; d^2_{1,1} \cdots d^2_{1,N^2_1-1} \quad d^2_{2,0} \; d^2_{2,1} \cdots d^2_{2,N^2_2-1} \quad d^2_{3,0} \; d^2_{3,1} \cdots d^2_{3,N^2_3-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$
$$d^0_{0,0} \quad d^0_{0,1} \quad d^0_{0,2} \quad \cdots \quad d^0_{0,N^0_0-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$
$$d^1_{0,0} \quad d^1_{0,1} \quad d^1_{0,2} \quad \cdots \quad d^1_{0,N^1_0-1} \qquad d^1_{1,0} \quad d^1_{1,1} \quad d^1_{1,2} \quad \cdots \quad d^1_{1,N^1_1-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{0,0}\, \boldsymbol{\phi}^2_{0,1} \cdots \boldsymbol{\phi}^2_{0,N^2_0-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{1,0}\, \boldsymbol{\phi}^2_{1,1} \cdots \boldsymbol{\phi}^2_{1,N^2_1-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{2,0}\, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^2_{3,0}\, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$
$$d^2_{0,0}\; d^2_{0,1} \cdots d^2_{0,N^2_0-1} \quad d^2_{1,0}\; d^2_{1,1} \cdots d^2_{1,N^2_1-1} \quad d^2_{2,0}\; d^2_{2,1} \cdots d^2_{2,N^2_2-1} \quad d^2_{3,0}\; d^2_{3,1} \cdots d^2_{3,N^2_3-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$
$$d^0_{0,0} \quad d^0_{0,1} \quad d^0_{0,2} \quad \cdots \quad d^0_{0,N^0_0-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$
$$d^1_{0,0} \quad d^1_{0,1} \quad d^1_{0,2} \quad \cdots \quad d^1_{0,N^1_0-1} \qquad d^1_{1,0} \quad d^1_{1,1} \quad d^1_{1,2} \quad \cdots \quad d^1_{1,N^1_1-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{2,0} \, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} \, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$
$$d^2_{2,0} \; d^2_{2,1} \cdots d^2_{2,N^2_2-1} \qquad d^2_{3,0} \; d^2_{3,1} \cdots d^2_{3,N^2_3-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$

$$d^0_{0,0} \quad d^0_{0,1} \quad d^0_{0,2} \quad \cdots \quad d^0_{0,N^0_0-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^1_{1,0} & \boldsymbol{\phi}^1_{1,1} & \boldsymbol{\phi}^1_{1,2} & \cdots & \boldsymbol{\phi}^1_{1,N^1_1-1} \end{bmatrix}$$

$$d^1_{0,0} \quad d^1_{0,1} \quad d^1_{0,2} \quad \cdots \quad d^1_{0,N^1_0-1} \qquad d^1_{1,0} \quad d^1_{1,1} \quad d^1_{1,2} \quad \cdots \quad d^1_{1,N^1_1-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{2,0} \, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} \, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$

$$d^2_{2,0} \; d^2_{2,1} \cdots d^2_{2,N^2_2-1} \qquad d^2_{3,0} \; d^2_{3,1} \cdots d^2_{3,N^2_3-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$
$$d^0_{0,0} \quad d^0_{0,1} \quad d^0_{0,2} \quad \cdots \quad d^0_{0,N^0_0-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix}$$
$$d^1_{0,0} \quad d^1_{0,1} \quad d^1_{0,2} \quad \cdots \quad d^1_{0,N^1_0-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{2,0} \, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} \, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$
$$d^2_{2,0} \; d^2_{2,1} \cdots d^2_{2,N^2_2-1} \qquad d^2_{3,0} \; d^2_{3,1} \cdots d^2_{3,N^2_3-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^0_{0,0} & \boldsymbol{\phi}^0_{0,1} & \boldsymbol{\phi}^0_{0,2} & \cdots & \boldsymbol{\phi}^0_{0,N^0_0-1} \end{bmatrix}$$

$$d^0_{0,0} \quad d^0_{0,1} \quad d^0_{0,2} \quad \cdots \quad d^0_{0,N^0_0-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix}$$

$$d^1_{0,0} \quad d^1_{0,1} \quad d^1_{0,2} \quad \cdots \quad d^1_{0,N^1_0-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{2,0} \, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} \, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$

$$d^2_{2,0} \; d^2_{2,1} \cdots d^2_{2,N^2_2-1} \qquad d^2_{3,0} \; d^2_{3,1} \cdots d^2_{3,N^2_3-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix}$$
$$d^1_{0,0} \quad d^1_{0,1} \quad d^1_{0,2} \quad \cdots \quad d^1_{0,N^1_0-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{2,0} \, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} \, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$
$$d^2_{2,0} \; d^2_{2,1} \cdots d^2_{2,N^2_2-1} \qquad d^2_{3,0} \; d^2_{3,1} \cdots d^2_{3,N^2_3-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix}$$
$$d^1_{0,0} \quad d^1_{0,1} \quad d^1_{0,2} \quad \cdots \quad d^1_{0,N^1_0-1}$$

$$\begin{bmatrix} \boldsymbol{\phi}^2_{2,0} \, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} \, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$
$$d^2_{2,0} \; d^2_{2,1} \cdots d^2_{2,N^2_2-1} \qquad d^2_{3,0} \; d^2_{3,1} \cdots d^2_{3,N^2_3-1}$$

According to cost functional $\mathscr{J}$, this is the best basis for approximation.

$$\begin{bmatrix} \boldsymbol{\phi}^1_{0,0} & \boldsymbol{\phi}^1_{0,1} & \boldsymbol{\phi}^1_{0,2} & \cdots & \boldsymbol{\phi}^1_{0,N^1_0-1} \end{bmatrix}$$
$$d^1_{0,0} \quad d^1_{0,1} \quad d^1_{0,2} \quad \cdots \quad d^1_{0,N^1_0-1}$$

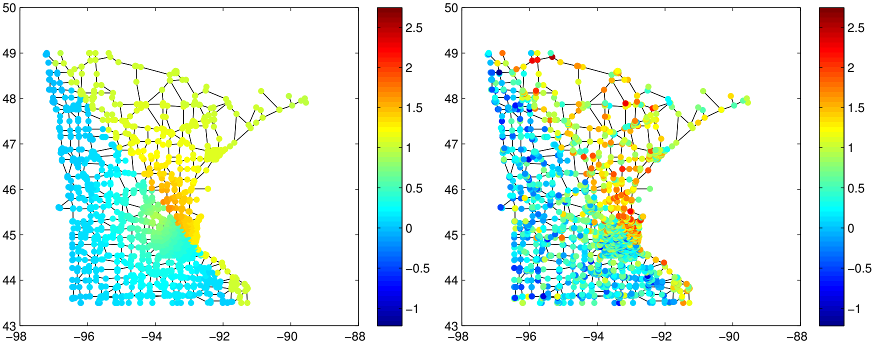
$$\begin{bmatrix} \boldsymbol{\phi}^2_{2,0} \, \boldsymbol{\phi}^2_{2,1} \cdots \boldsymbol{\phi}^2_{2,N^2_2-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\phi}^2_{3,0} \, \boldsymbol{\phi}^2_{3,1} \cdots \boldsymbol{\phi}^2_{3,N^2_3-1} \end{bmatrix}$$
$$d^2_{2,0} \; d^2_{2,1} \cdots d^2_{2,N^2_2-1} \qquad d^2_{3,0} \; d^2_{3,1} \cdots d^2_{3,N^2_3-1}$$

According to cost functional $\mathscr{J}$, this is the best basis for approximation.

- With the GHWT bases, we run the best-basis algorithm on both the default (coarse-to-fine) dictionary and the reorganized (fine-to-coarse) dictionary and then compare the cost of the 2 bases to determine the best-basis.

Outline

Original Signal vs. Noisy Signal



(a) Original signal: mutilated Gaussian

(b) Noisy signal: SNR = 5dB

Denoising Algorithm

1. Perform the HGLET / GHWT on the noisy signal
2. Run the best-basis algorithm
3. Soft-threshold and find the fraction of coefficients kept that yields the highest SNR
   - Sort the best-basis coefficients in non-increasing order of magnitude
   - Specify a magnitude threshold, T
   - Soft-threshold the coefficients d:

$$d_{ST}(i) = \begin{cases} \operatorname{sign}(d(i)) \cdot (|d(i)| - T) & \text{if } |d(i)| > T \\ 0 & \text{otherwise} \end{cases}$$

   - Note: keep all scaling coefficients intact

Denoising Algorithm

1. Perform the HGLET / GHWT on the noisy signal
2. Run the best-basis algorithm
3. Soft-threshold and find the fraction of coefficients kept that yields the highest SNR
   - Sort the best-basis coefficients in non-increasing order of magnitude
   - Specify a magnitude threshold, T
   - Soft-threshold the coefficients d:

$$d_{ST}(i) = \begin{cases} \mathrm{sign}(d(i)) \cdot (|d(i)| - T) & \text{if } |d(i)| > T \\ 0 & \text{otherwise} \end{cases}$$

   - Note: keep all scaling coefficients intact

Denoising Algorithm

1. Perform the HGLET / GHWT on the noisy signal
2. Run the best-basis algorithm
3. Soft-threshold and find the fraction of coefficients kept that yields the highest SNR
    - Sort the best-basis coefficients in non-increasing order of magnitude
    - Specify a magnitude threshold, T
    - Soft-threshold the coefficients d:

$$d_{\mathsf{ST}}(i) = \begin{cases} \operatorname{sign}(d(i)) \cdot (|d(i)| - T) & \text{if } |d(i)| > T \\ 0 & \text{otherwise} \end{cases}$$

    - Note: keep all scaling coefficients intact

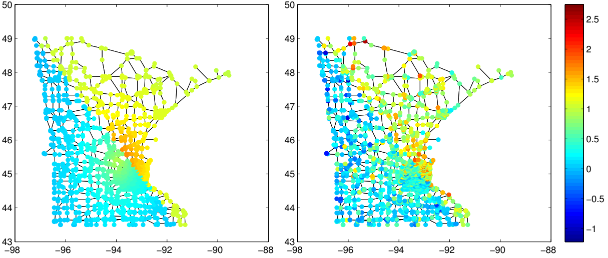
Denoising Algorithm

1. Perform the HGLET / GHWT on the noisy signal
2. Run the best-basis algorithm
3. Soft-threshold and find the fraction of coefficients kept that yields the highest SNR
   - Sort the best-basis coefficients in non-increasing order of magnitude
   - Specify a magnitude threshold, T
   - Soft-threshold the coefficients d:

$$d_{\mathsf{ST}}(i) = \begin{cases} \operatorname{sign}(d(i)) \cdot (|d(i)| - T) & \text{if } |d(i)| > T \\ 0 & \text{otherwise} \end{cases}$$

   - Note: keep all scaling coefficients intact

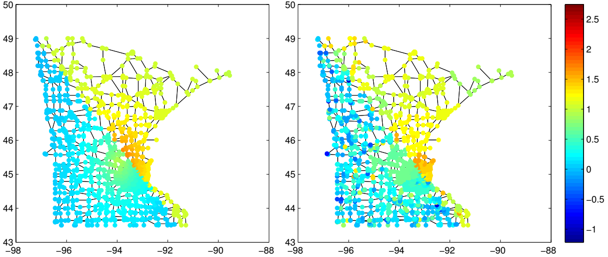
Results



(a) Original (b) Denoised via HGLET

Transform	% Coefficients Kept	SNR
HGLET	**49%**	**6.77 dB**

Results

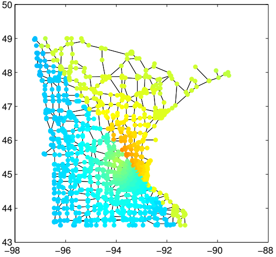


(a) Original

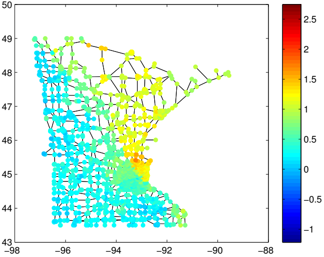
(c) Denoised via GHWT (coarse-to-fine)

Transform	% Coefficients Kept	SNR
HGLET	49%	6.77 dB
GHWT (coarse-to-fine)	**25%**	**10.11 dB**

Results



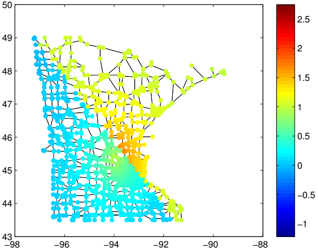
(a) Original



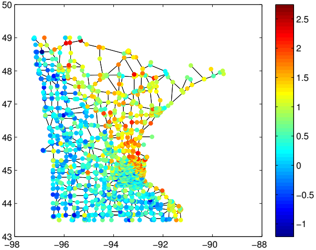
(d) Denoised via GHWT (fine-to-coarse)

Transform	% Coefficients Kept	SNR
HGLET	49%	6.77 dB
GHWT (coarse-to-fine)	25%	10.11 dB
GHWT (fine-to-coarse)	**11%**	**11.56 dB**

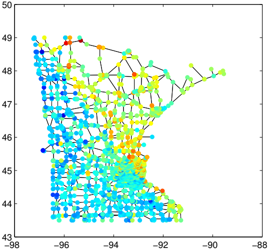
Results



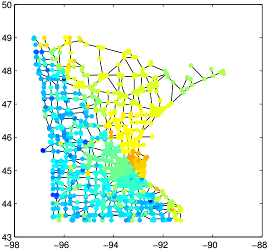
(a) Original



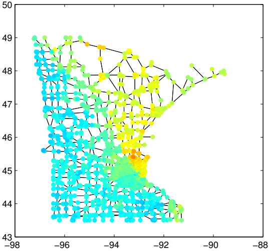
(b) Noisy signal



(c) HGLET



(d) GHWT (coarse-to-fine)



(e) GHWT (fine-to-coarse)

Observations

Transform	% Coefficients Kept	SNR
HGLET	49%	6.77 dB
GHWT (coarse-to-fine)	25%	10.11 dB
GHWT (fine-to-coarse)	11%	11.56 dB

- Even though its basis vectors are piecewise-constant, the GHWT does a good job of reconstructing the (mostly) smooth signal
  - It outperforms the HGLET, which has basis vectors that are smooth on their support
- The GHWT fine-to-coarse best basis achieved a higher SNR than the coarse-to-fine best basis
  - This suggests that for the purpose of denoising, grouping basis vectors/coefficients by 'frequency' is more effective than grouping by location

Outline

Summary: Harmonic Analysis of/on Irregular Domains via Laplacian Eigenfunctions

- LEs computed via the commuting integral operator provide an orthonormal basis on a general shape domain or a graph and allow spectral analysis/synthesis of data on them
- Can get fast-decaying expansion coefficients thanks to the rather implicit BC that may be more natural under certain situations
- Can decouple *geometry* of domains and *statistics* of data
- Can extract geometric information of a domain via $\{\lambda_k\}_k$
- Allow object-oriented (or localized) data analysis & synthesis, e.g., could be effective for local reconstruction of an ROI and anomaly detection on it
- ∃ A variety of applications: interpolation, extrapolation, local feature computation, solving heat equations on complicated domains ...
- Fast algorithms are the key for higher dimensions/large domains
- Can also be defined and computed on a *Riemannian manifold* (e.g., a curved surface); to do so, we need the *Riemannian metric* of the manifold and *geodesic distances* between sample points

Summary: Harmonic Analysis of/on Graphs via Laplacian Eigenfunctions

- Although graph Laplacian eigenvectors have been popular as replacement of the Fourier basis on a graph, the analogy takes us only so far due to their sensitivity to the geometry and topology of underlying graphs.
- We developed multiscale basis dictionaries on graphs and networks: *HGLET* and *GHWT*. We also developed a corresponding *best-basis algorithm*.
- The HGLET is a generalization of *Hierarchical Block Discrete Cosine Transforms* originally developed for regularly-sampled signals and images.
- The GHWT is a generalization of the *Haar Transform* and the *Walsh-Hadamard Transform*.
- Both of these transforms allow us to choose an orthonormal basis suitable for the task at hand, e.g., approximation, classification, regression, . . .
- They may also be useful for regularly-sampled signals, e.g., can deal with signals of non-dyadic length; adaptive segmentation, . . .
- Developing harmonic analysis tools for directed graphs will be challenging $\Longrightarrow$ our idea: use *distance matrix + SVD* instead; to be continued!
- Connect to lots of interesting mathematics and applications: *harmonic analysis, discrete mathematics, mathematical physics, PDEs, differential geometry, signal & image processing, statistics, . . .*

References

Laplacian Eigenfunction Resource Page
http://www.math.ucdavis.edu/~saito/lapeig/ contains:

- My Course Note (elementary) on "Laplacian Eigenfunctions: Theory, Applications, and Computations"
- My Course Slides on "Harmonic Analysis on Graphs and Networks"
- Talk slides of the minisymposia on Laplacian Eigenfunctions at: ICIAM 2007, Zürich (Organizers: NS, Mauro Maggioni); SIAM Imaging Science Conference 2008, San Diego (Organizers: NS, Xiaomin Huo); IPAM 5-day Workshop 2009, UCLA (Organizers: Peter Jones, Denis Grebenkov, NS); SIAM Annual Meeting 2013, San Diego (Organizers: Chiu-Yen Kao, Braxton Osting, NS).

The following articles (and the other related ones) are available at
http://www.math.ucdavis.edu/~saito/publications/

- N. Saito & J.-F. Remy: "The polyharmonic local sine transform: A new tool for local image analysis and synthesis without edge effect," *Applied & Computational Harmonic Analysis*, vol. 20, no. 1, pp. 41-73, 2006.
- N. Saito: "Data analysis and representation using eigenfunctions of Laplacian on a general domain," *Applied & Computational Harmonic Analysis*, vol. 25, no. 1, pp. 68–97, 2008.
- N. Saito & E. Woei: "Analysis of neuronal dendrite patterns using eigenvalues of graph Laplacians," *Japan SIAM Letters*, vol. 1, pp. 13–16, 2009.
- Y. Nakatsukasa, N. Saito, & E. Woei: "Mysteries around graph Laplacian eigenvalue 4," *Linear Algebra & Its Applications*, vol. 438, no. 8, pp. 3231–3246, 2013.
- J. Irion & N. Saito: "Hierarchical graph Laplacian eigen transforms," *Japan SIAM Letters*, vol. 6, pp. 21–24, 2014.
- J. Irion & N. Saito: "The generalized Haar-Walsh transform," *Proc. 2014 IEEE Workshop on Statistical Signal Processing*, pp. 488-491, 2014.

Thank you very much for your attention!