Laplacian Eigenfunctions: Foundations and Applications

Naoki Saito

Department of Mathematics
University of California, Davis

The Graduate University for Advanced Studies (SOKENDAI)
National Institute for Fusion Science
Toki-city, Gifu, Japan
September 4, 2013

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)
- Naofumi Iwama & Yoshio Nagayama (NIFS/SOKENDAI)
- Chiu-Yen Kao & Hrushikesh Mhaskar (Claremont Grad. Univ.)
- Braxton Osting (UCLA)
- Martin Reuter (Harvard Medical School)
- My current & former students at UC Davis
- Support from NSF & ONR
- The MacTutor History of Mathematics Archive, Wikipedia, ...

Outline

Lecture Outline

- Motivations
- History of Laplacian Eigenvalue Problems – Spectral Geometry
- Some Computational Procedures for Laplacian Eigenvalue Problems
- Lunch Break
- Laplacian Eigenfunctions via Commuting Integral Operator
- Applications
- Laplacian Eigenvalue Problems on Graphs
- Summary

General References

- H. Urakawa: *Laplacian & Networks*, Shokabo, 1996 (in Japanese).
- S. Kotani & H. Matano: *Differential Equations & Eigenfunction Expansions*, Iwanami, 2006 (in Japanese).
- 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.
- F. R. K. Chung: *Spectral Graph Theory*, Amer. Math. Soc., 1997.
- D. S. Grebenkov & B.-T. Nguyen: "Geometrical structure of Laplacian eigenfunctions," to appear in *SIAM Review*, 2013 (available as ArXiv:1206.1278v2 [math.AP]).
- Specific references are given within the lectures.
- Visit

  `http://www.math.ucdavis.edu/~saito/courses/LapEig/refs.html`

  and

  `http://www.math.ucdavis.edu/~saito/courses/HarmGraph/refs.html`

Outline

Motivations

- 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. $\Longrightarrow$ fast decaying expansion coefficients relative to a meaningful basis.
- Want to extract geometric information about the domain $\Omega \Longrightarrow$ shape clustering/classification.

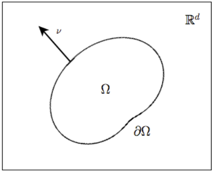


Figure: $\Omega \subset \mathbb{R}^d$ with ν being a normal vector on $\partial\Omega$.

Motivations

- 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. $\implies$ fast decaying expansion coefficients relative to a meaningful basis.
- Want to extract geometric information about the domain $\Omega \implies$ shape clustering/classification.

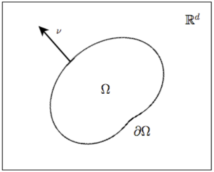


Figure: $\Omega \subset \mathbb{R}^d$ with ν being a normal vector on $\partial\Omega$.

Motivations

- 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. $\implies$ fast decaying expansion coefficients relative to a meaningful basis.
- Want to extract geometric information about the domain $\Omega \implies$ shape clustering/classification.

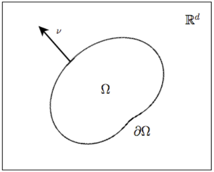


Figure: $\Omega \subset \mathbb{R}^d$ with ν being a normal vector on $\partial\Omega$.

Motivations

- 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. $\implies$ fast decaying expansion coefficients relative to a meaningful basis.
- Want to extract geometric information about the domain $\Omega \implies$ shape clustering/classification.

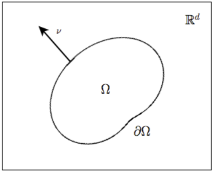
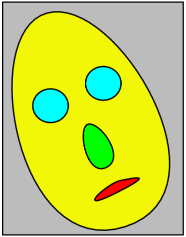
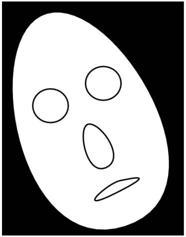


Figure: $\Omega \subset \mathbb{R}^d$ with ν being a normal vector on $\partial\Omega$.

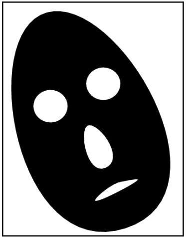
Object-Oriented Image Analysis



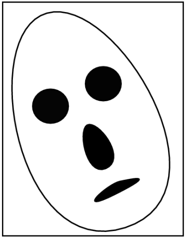
(a) Original



(b) Background

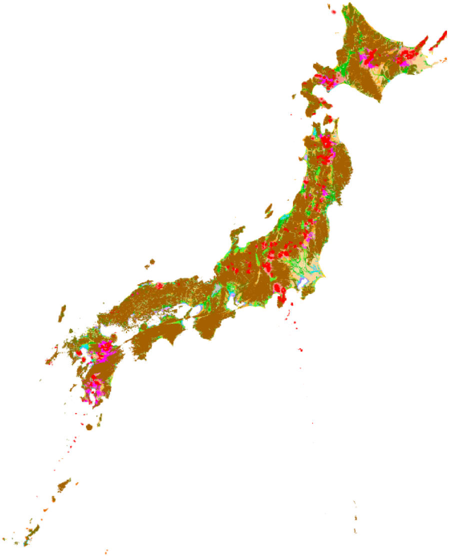


(c) Object

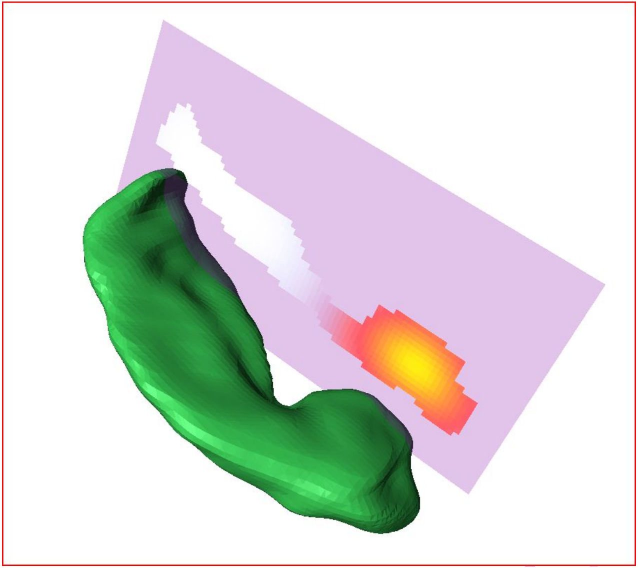


(d) Anomalies

Data Analysis on a Complicated Domain



3D Hippocampus Shape Analysis



Enter Laplacian Eigenfunctions!

- Consider a domain $\Omega \subset \mathbb{R}^d$ of general shape.
- Let $\mathscr{L} := -\Delta = -\left(\dfrac{\partial^2}{\partial {x_1}^2} + \cdots + \dfrac{\partial^2}{\partial {x_d}^2}\right)$.
- 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:* $\dfrac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$;
  - *Robin (or impedance):* $au + b\dfrac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$, $a \neq 0 \neq b$.

Enter Laplacian Eigenfunctions!

- Consider a domain $\Omega \subset \mathbb{R}^d$ of general shape.
- 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!

- Consider a domain $\Omega \subset \mathbb{R}^d$ of general shape.
- 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!

- Consider a domain $\Omega \subset \mathbb{R}^d$ of general shape.
- 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!

- Consider a domain $\Omega \subset \mathbb{R}^d$ of general shape.
- 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!

- Consider a domain $\Omega \subset \mathbb{R}^d$ of general shape.
- 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!

- Consider a domain $\Omega \subset \mathbb{R}^d$ of general shape.
- 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.
- We know that in the case of the Dirichlet BC $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k \to \infty$.
- 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.
- We know that in the case of the Dirichlet BC $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k \to \infty$.
- 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.
- We know that in the case of the Dirichlet BC $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k \to \infty$.
- 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 . . .

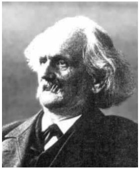
- The nontrivial solution $u = \varphi$ of such a *boundary value problem* (BVP) is called the Laplacian eigenfunction corresponding to the eigenvalue λ.
- We know that in the case of the Dirichlet BC
  $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k \to \infty$.
- On the other hand, the Neumann 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.
- We know that in the case of the Dirichlet BC
  $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k \to \infty$.
- 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) Lejeune 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 a specific domain Ω using genuine basis functions tailored to the domain instead of the basis functions developed for rectangles, torus, intervals, etc.?
- After all, *sines* (and *cosines*) are the eigenfunctions of the Laplacian on the *rectangular* domain 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 a specific domain Ω using genuine basis functions tailored to the domain instead of the basis functions developed for rectangles, torus, intervals, etc.?
- After all, *sines* (and *cosines*) are the eigenfunctions of the Laplacian on the *rectangular* domain 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 a specific domain Ω using genuine basis functions tailored to the domain instead of the basis functions developed for rectangles, torus, intervals, etc.?
- After all, *sines* (and *cosines*) are the eigenfunctions of the Laplacian on the *rectangular* domain 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 a specific domain Ω using genuine basis functions tailored to the domain instead of the basis functions developed for rectangles, torus, intervals, etc.?
- After all, *sines* (and *cosines*) are the eigenfunctions of the Laplacian on the *rectangular* domain 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. (More about this later.)
- 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. (More about this later.)
- 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. (More about this later.)
- 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): $\rho(x)\dfrac{\partial^2 u}{\partial t^2} = \dfrac{\partial}{\partial x}\left(T(x)\dfrac{\partial u}{\partial x}\right)$

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): $\rho(x)\dfrac{\partial^2 u}{\partial t^2} = \dfrac{\partial}{\partial x}\left(T(x)\dfrac{\partial u}{\partial x}\right)$

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: $u(0,t) = u(\ell,t) = 0$, $\forall t \ge 0$.
- 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:

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

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:

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

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:

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

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.

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

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

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., $r = \pm \mathrm{i}\xi$. 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

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

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., $r = \pm\mathrm{i}\xi$. 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 $T'' = -\xi^2c^2T$ 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 $T'' = -\xi^2c^2T$ 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

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

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

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

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

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

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

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

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

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

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,

$$\begin{aligned} \text{small } \lambda_1 &\iff \text{long } \ell \\ \text{large } \lambda_1 &\iff \text{short } \ell \end{aligned}$$

- 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 $0 < \lambda_1 \le \lambda_2 \le \lambda_3 \le \cdots \le \lambda_k \le \cdots \to \infty$ 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 $0 < \lambda_1 \le \lambda_2 \le \lambda_3 \le \cdots \le \lambda_k \le \cdots \to \infty$ 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 . . .

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):

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

(a) Hermann Weyl (1885–1955)

(b) Subbaramiah Minakshisundaram (1913–1968)

(c) Åke Pleijel (1913–1989)

(d) Mark Kac (1914–1984)

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

- $\lambda_{m+1} - \lambda_m \le 2 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j\,; \quad \lambda_{m+1} \le 3 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j\,; \quad \frac{\lambda_{m+1}}{\lambda_m} \le 3.$
- $\sum_{j=1}^{m} \frac{\lambda_j}{\lambda_{m+1} - \lambda_j} \ge \frac{m}{2}$ (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) George Pólya (1887–1985)

(c) Hans Weinberger (1928–)

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

- $\lambda_{m+1} - \lambda_m \le 2 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j$; $\lambda_{m+1} \le 3 \cdot \frac{1}{m} \sum_{j=1}^{m} \lambda_j$; $\frac{\lambda_{m+1}}{\lambda_m} \le 3$.
- $\sum_{j=1}^{m} \frac{\lambda_j}{\lambda_{m+1} - \lambda_j} \ge \frac{m}{2}$ (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) George Pólya (1887–1985)

(c) Hans Weinberger (1928–)

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) George Pólya (1887–1985)

(c) Hans 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 $\sqrt{\alpha^2 + \beta^2} \times \sqrt{(a-\alpha)^2 + (b-\beta)^2}$:

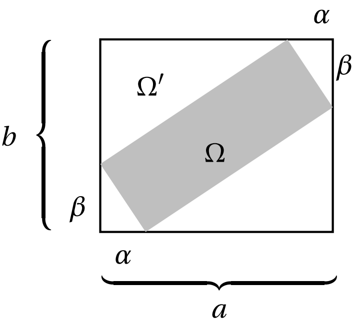


Figure: The Neumann BC generates an counterexample (From A. Henrot, 2006)

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

Method of Particular Solutions (MPS)

- Initiated by Fox, Henrich, Moler (1967)

- Below, I will explain the basics of MPS using the slides created by Prof. Alex Barnett of Dartmouth College.
- For the details, see:
  - T. Betcke & L. N. Trefethen: "Reviving the method of particular solutions," *SIAM Review*, vol. 47, no. 3, pp. 469–491, 2005.
  - A. H. Barnett & T. Betcke: "Quantum mushroom billiards," *Chaos*, no. 4, 043125, 13 pp., 2007.
- The MATLAB-based software package is available at:

  `http://code.google.com/p/mpspack`

Method of Particular Solutions (MPS)

- Initiated by Fox, Henrich, Moler (1967)

Figure: The MATLAB® Logo!

- Below, I will explain the basics of MPS using the slides created by Prof. Alex Barnett of Dartmouth College.
- For the details, see:
  - T. Betcke & L. N. Trefethen: "Reviving the method of particular solutions," *SIAM Review*, vol. 47, no. 3, pp. 469–491, 2005.
  - A. H. Barnett & T. Betcke: "Quantum mushroom billiards," *Chaos*, no. 4, 043125, 13 pp., 2007.
- The MATLAB-based software package is available at:

`http://code.google.com/p/mpspack`

Method of Particular Solutions (MPS)

- Initiated by Fox, Henrich, Moler (1967)

Figure: The MATLAB® Logo!

- Below, I will explain the basics of MPS using the slides created by Prof. Alex Barnett of Dartmouth College.
- For the details, see:
  - T. Betcke & L. N. Trefethen: "Reviving the method of particular solutions," *SIAM Review*, vol. 47, no. 3, pp. 469–491, 2005.
  - A. H. Barnett & T. Betcke: "Quantum mushroom billiards," *Chaos*, no. 4, 043125, 13 pp., 2007.
- The MATLAB-based software package is available at:

  http://code.google.com/p/mpspack

Method of Particular Solutions (MPS)

- Initiated by Fox, Henrich, Moler (1967)

Figure: The MATLAB® Logo!

- Below, I will explain the basics of MPS using the slides created by Prof. Alex Barnett of Dartmouth College.
- For the details, see:
  - T. Betcke & L. N. Trefethen: "Reviving the method of particular solutions," *SIAM Review*, vol. 47, no. 3, pp. 469–491, 2005.
  - A. H. Barnett & T. Betcke: "Quantum mushroom billiards," *Chaos*, no. 4, 043125, 13 pp., 2007.
- The MATLAB-based software package is available at:

  `http://code.google.com/p/mpspack`

Method of Particular Solutions (MPS)

- Initiated by Fox, Henrich, Moler (1967)

Figure: The MATLAB® Logo!

- Below, I will explain the basics of MPS using the slides created by Prof. Alex Barnett of Dartmouth College.
- For the details, see:
  - T. Betcke & L. N. Trefethen: "Reviving the method of particular solutions," *SIAM Review*, vol. 47, no. 3, pp. 469–491, 2005.
  - A. H. Barnett & T. Betcke: "Quantum mushroom billiards," *Chaos*, no. 4, 043125, 13 pp., 2007.
- The MATLAB-based software package is available at:

  `http://code.google.com/p/mpspack`

Method of Particular Solutions (MPS)

- Initiated by Fox, Henrich, Moler (1967)

Figure: The MATLAB® Logo!

- Below, I will explain the basics of MPS using the slides created by Prof. Alex Barnett of Dartmouth College.
- For the details, see:
  - T. Betcke & L. N. Trefethen: "Reviving the method of particular solutions," *SIAM Review*, vol. 47, no. 3, pp. 469–491, 2005.
  - A. H. Barnett & T. Betcke: "Quantum mushroom billiards," *Chaos*, no. 4, 043125, 13 pp., 2007.
- The MATLAB-based software package is available at:

  `http://code.google.com/p/mpspack`

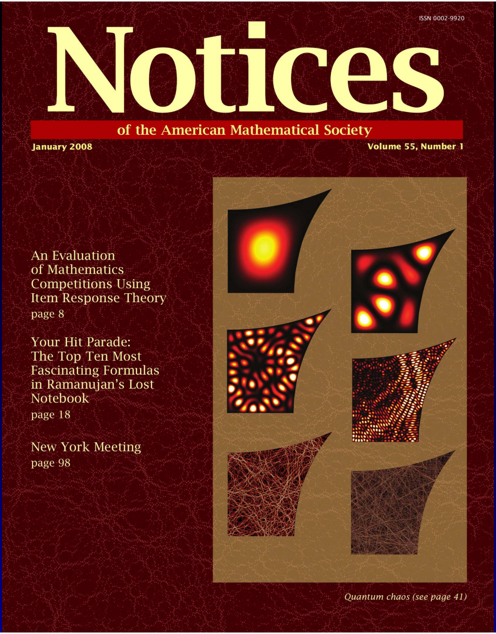
Numerical test

eigenfunctions ϕ_j for $j = 1, 10, 10^2, 10^3, 10^4, 10^5$

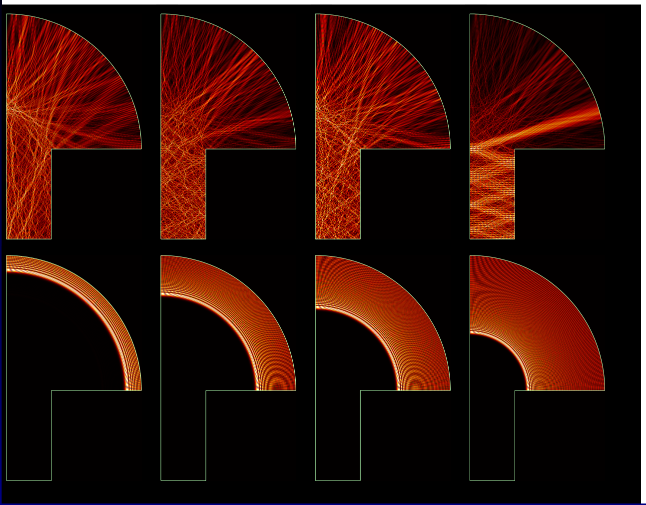
background:
random plane waves,
a model for modes
(Berry '77)

tested 30000 ϕ_j's: strong evidence for QUE (B '06)

How compute many ϕ_j efficiently to $j \sim 10^6$, 10^3 wavelengths across?



High freq. mushroom eigenfunctions



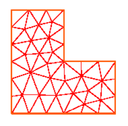
- $j \approx 5 \times 10^4$, 20 sec per mode (bdry data only; longer for interior)

Two classes of numerical methods for eigenmodes

A) Volume discretization

finite differencing

finite element (hp-FEM)



- local basis representation

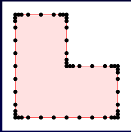
  e.g. polynomials in elements
- basis satisfies BCs, not the PDE
- basis size $N \geq O(k^d)$

  "pollution" (Babuska–Sauter)
- $k_j^2 \approx$ sparse matrix eigenvalues

B) Boundary discretization

boundary integral equations (BIE)

method of particular solutions (MPS)



- global basis representation

  e.g. layer potentials, plane waves
- basis satisfies PDE $-\Delta u = k^2 u$
- basis size $N = O(k^{d-1})$

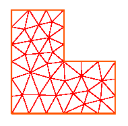
  e.g. factor 10^3 smaller
- dense nonlinear eigenval. prob.

Two classes of numerical methods for eigenmodes

A) Volume discretization

finite differencing

finite element (hp-FEM)



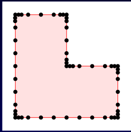
- local basis representation
  e.g. polynomials in elements
- basis satisfies BCs, not the PDE
- basis size $N \geq O(k^d)$
  "pollution" (Babuska–Sauter)
- $k_j^2 \approx$ sparse matrix eigenvalues

$j < 10^4$, rel. err. 10^{-3} (Heuveline)

B) Boundary discretization

boundary integral equations (BIE)

method of particular solutions (MPS)



- global basis representation
  e.g. layer potentials, plane waves
- basis satisfies PDE $-\Delta u = k^2 u$
- basis size $N = O(k^{d-1})$
  e.g. factor 10^3 smaller
- dense nonlinear eigenval. prob.

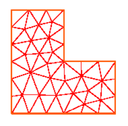
$j > 10^6$, err. 10^{-14} (Tureci, B–Hassell)

Two classes of numerical methods for eigenmodes

A) Volume discretization

finite differencing

finite element (hp-FEM)



- local basis representation

  e.g. polynomials in elements
- basis satisfies BCs, not the PDE
- basis size $N \geq O(k^d)$

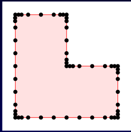
  "pollution" (Babuska–Sauter)
- $k_j^2 \approx$ sparse matrix eigenvalues

$j < 10^4$, rel. err. 10^{-3} (Heuveline)

B) Boundary discretization

boundary integral equations (BIE)

method of particular solutions (MPS)



- global basis representation

  e.g. layer potentials, plane waves
- basis satisfies PDE $-\Delta u = k^2 u$
- basis size $N = O(k^{d-1})$

  e.g. factor 10^3 smaller
- dense nonlinear eigenval. prob.

$j > 10^6$, err. 10^{-14} (Tureci, B–Hassell)

$\Rightarrow$ boundary methods much more powerful, but nonlinearity an issue

II. Numerics: global basis methods

Want nontriv. solns to $(\Delta + E)u = 0$ in Ω Helmholtz

$u = 0$ on $\partial\Omega$

II. Numerics: global basis methods

Want nontriv. solns to $(\Delta + E)u = 0$ in Ω — Helmholtz
$u = 0$ on $\partial\Omega$

Guess energy E, trial func. $u(\mathbf{x}) \approx \sum_{l=1}^{N} \alpha_l \xi_l(\mathbf{x})$, $(\Delta + E)\xi_l = 0$ in Ω

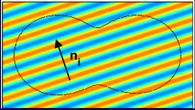
– p. 13

II. Numerics: global basis methods

Want nontriv. solns to $(\Delta + E)u = 0$ in Ω; $u = 0$ on $\partial\Omega$ — Helmholtz

Guess energy E, trial func. $u(\mathbf{x}) \approx \sum_{l=1}^{N} \alpha_l \xi_l(\mathbf{x})$, $(\Delta + E)\xi_l = 0$ in Ω

Need basis $\{\xi_l\}$ to well approximate eigenfunctions, e.g. . .



Plane waves $\sin(k\boldsymbol{n}_l \cdot \mathbf{x})$, $k^2 = E$

Fourier-Bessel $J_l(kr)\sin(l\theta)$

Thm: Ω analytic $\Rightarrow$ exponential convergence (Eisenstat '74)

i.e. best error in $u = O(c^{-N})$

c = conformal dist. from $\partial\Omega$ to nearest singularity in analytic continuation of u

– p. 13

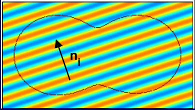
II. Numerics: global basis methods

Want nontriv. solns to $(\Delta + E)u = 0$ in Ω, $u = 0$ on $\partial\Omega$ — Helmholtz

$$(\Delta + E)u = 0 \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \partial\Omega$$

Guess energy E, trial func. $u(\mathbf{x}) \approx \sum_{l=1}^{N} \alpha_l \xi_l(\mathbf{x})$, $\quad (\Delta + E)\xi_l = 0$ in Ω

Need basis $\{\xi_l\}$ to well approximate eigenfunctions, e.g. . .



Plane waves	$\sin(k\boldsymbol{n}_l \cdot \mathbf{x})$,	$k^2 = E$
Fourier-Bessel	$J_l(kr)\sin(l\theta)$	

Thm: Ω analytic $\Rightarrow$ exponential convergence (Eisenstat '74)

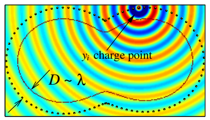
i.e. best error in $u = O(c^{-N})$

c = conformal dist. from $\partial\Omega$ to nearest singularity in analytic continuation of u

- Practice: usually fail! (coeff $\|\boldsymbol{\alpha}\|_2 \gg 10^{16}$ to achieve theorem)

Develop better bases for when singularities nearby or at corners . . .

More flexible global basis sets



Fundamental solutions (MFS):

$$H_0^{(1)}(k|\mathbf{x} - \boldsymbol{y}_l|), \text{ with } \{\boldsymbol{y}_l\} \text{ outside } \Omega$$

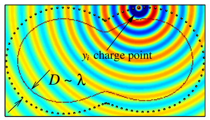
For Ω analytic and MFS lie on closed curve Γ:

Γ shields singularities in anal. cont. of $u \Leftrightarrow \|\boldsymbol{\alpha}\|_2 = O(1)$

(B-Betcke JCP '08)

Practice: excellent, including non-reentrant corners

More flexible global basis sets



Fundamental solutions (MFS):

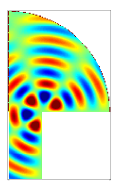
$H_0^{(1)}(k|\mathbf{x} - \boldsymbol{y}_l|)$, with $\{\boldsymbol{y}_l\}$ outside Ω

For Ω analytic and MFS lie on closed curve Γ:

Γ shields singularities in anal. cont. of $u \Leftrightarrow \|\boldsymbol{\alpha}\|_2 = O(1)$

(B-Betcke JCP '08)

Practice: excellent, including non-reentrant corners



Corner-adapted Fourier-Bessel:

$J_{\beta l}(kr)\sin(\beta l \theta)$

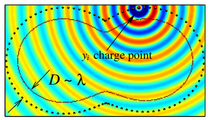
for singular corner $\theta = \pi/\beta$, β non-integer

Practice: exp. conv. for multiple corners (Betcke '05)

mushroom w/ scaling method (B-Betcke '07)

– p. 14

More flexible global basis sets

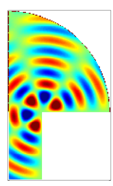


Fundamental solutions (MFS):

$H_0^{(1)}(k|\mathbf{x} - \boldsymbol{y}_l|)$, with $\{\boldsymbol{y}_l\}$ outside Ω

For Ω analytic and MFS lie on closed curve Γ:
Γ shields singularities in anal. cont. of u $\Leftrightarrow$ $\|\boldsymbol{\alpha}\|_2 = O(1)$
(B-Betcke JCP '08)

Practice: excellent, including non-reentrant corners



Corner-adapted Fourier-Bessel:

$J_{\beta l}(kr) \sin(\beta l \theta)$

for singular corner $\theta = \pi/\beta$, β non-integer

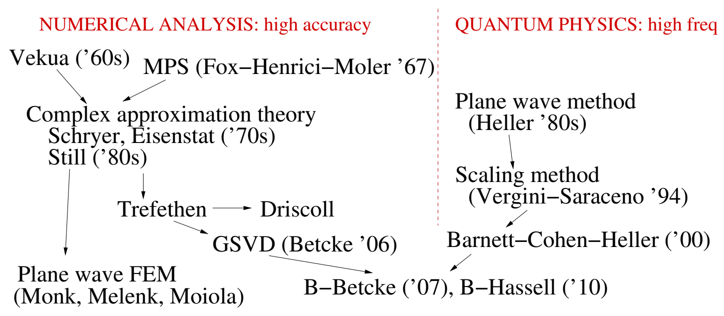
Practice: exp. conv. for multiple corners (Betcke '05)
mushroom w/ scaling method (B-Betcke '07)

All such global methods much better than FEM at large k: $N = O(k)$

- price to pay for high accuracy is understanding analyticity of u

History of global basis approximation

global bases a.k.a Method of particular solutions (MPS)



Recent weaving together of ideas from physics and numerical math. . .

Finding eigenpairs E_j, ϕ_j with the MPS

If u approximates ϕ_j then $\int_{\partial\Omega} |u|^2 ds$ small (Fox et al. ’67, Heller ’84)

Small compared to what?

Finding eigenpairs E_j, ϕ_j with the MPS

If u approximates ϕ_j then $\int_{\partial\Omega} |u|^2 ds$ small (Fox et al. '67, Heller '84)

Small compared to what? want interior norm $\int_\Omega |u|^2 d\mathbf{x} = 1$, so ...

$$\text{tension} \quad t[u] := \left(\frac{\int_{\partial\Omega} |u|^2 ds}{\int_\Omega |u|^2 d\mathbf{x}} \right)^{1/2} = \left(\frac{\boldsymbol{\alpha}^* F \boldsymbol{\alpha}}{\boldsymbol{\alpha}^* G \boldsymbol{\alpha}} \right)^{1/2}$$ (Betcke, Barnett, ...)

↖ inner prod. matrices of bases

– p. 16

Finding eigenpairs E_j, ϕ_j with the MPS

If u approximates ϕ_j then $\int_{\partial\Omega} |u|^2 ds$ small (Fox et al. '67, Heller '84)

Small compared to what? want interior norm $\int_\Omega |u|^2 d\mathbf{x} = 1$, so …

$$\text{tension} \quad t[u] := \left(\frac{\int_{\partial\Omega} |u|^2 ds}{\int_\Omega |u|^2 d\mathbf{x}} \right)^{1/2} = \left(\frac{\boldsymbol{\alpha}^* F \boldsymbol{\alpha}}{\boldsymbol{\alpha}^* G \boldsymbol{\alpha}} \right)^{1/2}$$ (Betcke, Barnett, …)

↖ inner prod. matrices of bases

Best tension at each E, $\quad t_m(E) := \min_{\boldsymbol{\alpha}} t[u] = \lambda_1(F, G)$

min. generalized eigenvalue

Finding eigenpairs E_j, ϕ_j with the MPS

If u approximates ϕ_j then $\int_{\partial\Omega} |u|^2 ds$ small (Fox et al. '67, Heller '84)

Small compared to what? want interior norm $\int_\Omega |u|^2 d\mathbf{x} = 1$, so …

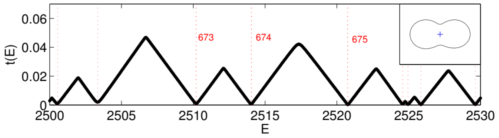
$$\text{tension} \quad t[u] := \left(\frac{\int_{\partial\Omega} |u|^2 ds}{\int_\Omega |u|^2 d\mathbf{x}} \right)^{1/2} = \left(\frac{\boldsymbol{\alpha}^* F \boldsymbol{\alpha}}{\boldsymbol{\alpha}^* G \boldsymbol{\alpha}} \right)^{1/2}$$

(Betcke, Barnett, …)

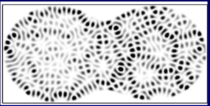
↖ inner prod. matrices of bases

Best tension at each E, $\quad t_m(E) := \min_{\boldsymbol{\alpha}} t[u] = \lambda_1(F, G)$

min. generalized eigenvalue



Ω analytic, $\quad j \approx 700$



Finding eigenpairs E_j, ϕ_j with the MPS

If u approximates ϕ_j then $\int_{\partial\Omega} |u|^2 ds$ small (Fox et al. '67, Heller '84)

Small compared to what? want interior norm $\int_\Omega |u|^2 d\mathbf{x} = 1$, so …

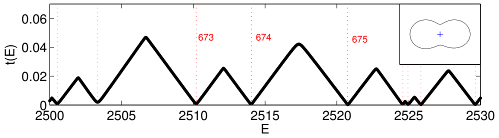
$$\text{tension} \quad t[u] := \left(\frac{\int_{\partial\Omega} |u|^2 ds}{\int_\Omega |u|^2 d\mathbf{x}} \right)^{1/2} = \left(\frac{\boldsymbol{\alpha}^* F \boldsymbol{\alpha}}{\boldsymbol{\alpha}^* G \boldsymbol{\alpha}} \right)^{1/2}$$

(Betcke, Barnett, …)

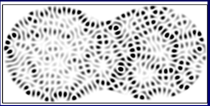
↖ inner prod. matrices of bases

Best tension at each E, $\quad t_m(E) := \min_{\boldsymbol{\alpha}} t[u] = \lambda_1(F, G)$

min. generalized eigenvalue



Ω analytic, $\quad j \approx 700$



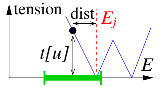
- iterative search along E axis: ~ 10 func. evals to find each min
- then eigenvector gives basis coeffs of approx. ϕ_j How accurate?

– p. 16

Numerical analysis: bounding errors

Say find small $t[u]$ at some E: how close is true E_j?

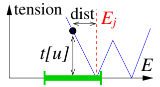
seek upper bound on $\mathrm{dist}(E, \mathrm{spec}) := \min_j |E_j - E|$



– p. 17

Numerical analysis: bounding errors

Say find small $t[u]$ at some E: how close is true E_j?

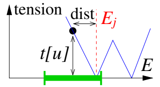


seek upper bound on $\text{dist}(E, \text{spec}) := \min_j |E_j - E|$

Thm (Moler-Payne '68): $\text{dist}(E, \text{spec}) \le C_{\text{MP}}\, E\, t[u]$

Noticed slopes of tension steeper than this at high E: can we beat MP?

Numerical analysis: bounding errors



Say find small $t[u]$ at some E: how close is true E_j?

seek upper bound on $\text{dist}(E, \text{spec}) := \min_j |E_j - E|$

Thm (Moler-Payne '68): $\text{dist}(E, \text{spec}) \le C_{\text{MP}}\, E\, t[u]$

Noticed slopes of tension steeper than this at high E: can we beat MP?

Thm (B '09, B-Hassell '10): $\text{dist}(E, \text{spec}) \le C_\Omega E^{1/2}\, t[u]$

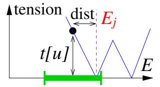
e.g $E = 10^6$ gives 10^3 better than MP: 3 extra digits for free!

best possible power of E; similar improvement for L^2-error of ϕ_j

Numerical analysis: bounding errors

Say find small $t[u]$ at some E: how close is true E_j?

seek upper bound on $\text{dist}(E, \text{spec}) := \min_j |E_j - E|$



Thm (Moler-Payne '68): $\quad \text{dist}(E, \text{spec}) \;\le\; C_{\text{MP}}\, E\, t[u]$

Noticed slopes of tension steeper than this at high E: can we beat MP?

Thm (B '09, B-Hassell '10): $\quad \text{dist}(E, \text{spec}) \;\le\; C_\Omega E^{1/2}\, t[u]$

e.g $E = 10^6$ gives 10^3 better than MP: 3 extra digits for free!
best possible power of E; similar improvement for L^2-error of ϕ_j

Proof: $\quad \exists\, E$-dep. bdry op. A s.t. $\quad \int_\Omega uv\, d\mathbf{x} \;=\; \int_{\partial\Omega} u(s)(Av)(s)ds$

$t[u]^{-2} \le \|A(E)\|_2$ which can bound via new quasi-orthogonality thm:

"all bdry funcs $\psi_j := \boldsymbol{n} \cdot \nabla\phi_j$ in semiclassical window are nearly orthog"

$$\left\| \sum_{|E_j - E| \le E^{1/2}} \psi_j \langle \psi_j, \cdot \rangle \right\|_2 \le C_\Omega E$$

norm of *each term* is $O(E)$,
Weyl says $O(E^{(d-1)/2})$ such terms

Example

Ω analytic

MFS (point charges) basis

$N = 500$

skip other details

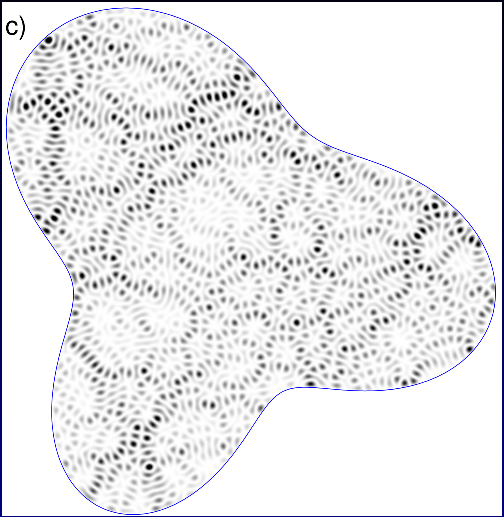
$t[u] = 2.2 \times 10^{-12}$ at

$E = 10005.0213579739$

Thm gives ± 3 in last digit

i.e. 14 digits accuracy

$j \approx 2552$



Outline

Method of Fundamental Solutions (MFS)

- Is highly efficient and accurate for computing Laplacian eigenvalues and eigenfunctions
- Can deal with singularities such as *corners* and *cracks* in a domain
- Is one of the *meshfree* methods; i.e., no meshing/gridding.
- Below, I will explain the basics of MFS using the slides created by Dr. Pedro Antunes of Univ. of Lisbon, Portugal.
- For the details, see:
  - C. J. S. Alves and P. R. S. Antunes: "The Method of Fundamental Solutions applied to the calculation of eigenfrequencies and eigenmodes of 2D simply connected shapes," *Computers, Materials & Continua*, vol. 2, no. 4, pp. 251–266, 2005.
  - P. R. S. Antunes: "Numerical calculation of eigensolutions of 3D shapes using the Method of Fundamental Solutions," *Numerical Methods for Partial Differential Equations*, vol. 27, no. 6, pp. 1525–1550, 2011.

Method of Fundamental Solutions (MFS)

- Is highly efficient and accurate for computing Laplacian eigenvalues and eigenfunctions
- Can deal with singularities such as *corners* and *cracks* in a domain
- Is one of the *meshfree* methods; i.e., no meshing/gridding.
- Below, I will explain the basics of MFS using the slides created by Dr. Pedro Antunes of Univ. of Lisbon, Portugal.
- For the details, see:
  - C. J. S. Alves and P. R. S. Antunes: "The Method of Fundamental Solutions applied to the calculation of eigenfrequencies and eigenmodes of 2D simply connected shapes," *Computers, Materials & Continua*, vol. 2, no. 4, pp. 251–266, 2005.
  - P. R. S. Antunes: "Numerical calculation of eigensolutions of 3D shapes using the Method of Fundamental Solutions," *Numerical Methods for Partial Differential Equations*, vol. 27, no. 6, pp. 1525–1550, 2011.

Method of Fundamental Solutions (MFS)

- Is highly efficient and accurate for computing Laplacian eigenvalues and eigenfunctions
- Can deal with singularities such as *corners* and *cracks* in a domain
- Is one of the *meshfree* methods; i.e., no meshing/gridding.
- Below, I will explain the basics of MFS using the slides created by Dr. Pedro Antunes of Univ. of Lisbon, Portugal.
- For the details, see:
  - C. J. S. Alves and P. R. S. Antunes: "The Method of Fundamental Solutions applied to the calculation of eigenfrequencies and eigenmodes of 2D simply connected shapes," *Computers, Materials & Continua*, vol. 2, no. 4, pp. 251–266, 2005.
  - P. R. S. Antunes: "Numerical calculation of eigensolutions of 3D shapes using the Method of Fundamental Solutions," *Numerical Methods for Partial Differential Equations*, vol. 27, no. 6, pp. 1525–1550, 2011.

Method of Fundamental Solutions (MFS)

- Is highly efficient and accurate for computing Laplacian eigenvalues and eigenfunctions
- Can deal with singularities such as *corners* and *cracks* in a domain
- Is one of the *meshfree* methods; i.e., no meshing/gridding.
- Below, I will explain the basics of MFS using the slides created by Dr. Pedro Antunes of Univ. of Lisbon, Portugal.
- For the details, see:
  - C. J. S. Alves and P. R. S. Antunes: "The Method of Fundamental Solutions applied to the calculation of eigenfrequencies and eigenmodes of 2D simply connected shapes," *Computers, Materials & Continua*, vol. 2, no. 4, pp. 251–266, 2005.
  - P. R. S. Antunes: "Numerical calculation of eigensolutions of 3D shapes using the Method of Fundamental Solutions," *Numerical Methods for Partial Differential Equations*, vol. 27, no. 6, pp. 1525–1550, 2011.

Method of Fundamental Solutions (MFS)

- Is highly efficient and accurate for computing Laplacian eigenvalues and eigenfunctions
- Can deal with singularities such as *corners* and *cracks* in a domain
- Is one of the *meshfree* methods; i.e., no meshing/gridding.
- Below, I will explain the basics of MFS using the slides created by Dr. Pedro Antunes of Univ. of Lisbon, Portugal.
- For the details, see:
  - C. J. S. Alves and P. R. S. Antunes: "The Method of Fundamental Solutions applied to the calculation of eigenfrequencies and eigenmodes of 2D simply connected shapes," *Computers, Materials & Continua*, vol. 2, no. 4, pp. 251–266, 2005.
  - P. R. S. Antunes: "Numerical calculation of eigensolutions of 3D shapes using the Method of Fundamental Solutions," *Numerical Methods for Partial Differential Equations*, vol. 27, no. 6, pp. 1525–1550, 2011.

Method of Fundamental Solutions (MFS)

- Is highly efficient and accurate for computing Laplacian eigenvalues and eigenfunctions
- Can deal with singularities such as *corners* and *cracks* in a domain
- Is one of the *meshfree* methods; i.e., no meshing/gridding.
- Below, I will explain the basics of MFS using the slides created by Dr. Pedro Antunes of Univ. of Lisbon, Portugal.
- For the details, see:
  - C. J. S. Alves and P. R. S. Antunes: "The Method of Fundamental Solutions applied to the calculation of eigenfrequencies and eigenmodes of 2D simply connected shapes," *Computers, Materials & Continua*, vol. 2, no. 4, pp. 251–266, 2005.
  - P. R. S. Antunes: "Numerical calculation of eigensolutions of 3D shapes using the Method of Fundamental Solutions," *Numerical Methods for Partial Differential Equations*, vol. 27, no. 6, pp. 1525–1550, 2011.

Method of Fundamental Solutions (MFS)

- Is highly efficient and accurate for computing Laplacian eigenvalues and eigenfunctions
- Can deal with singularities such as *corners* and *cracks* in a domain
- Is one of the *meshfree* methods; i.e., no meshing/gridding.
- Below, I will explain the basics of MFS using the slides created by Dr. Pedro Antunes of Univ. of Lisbon, Portugal.
- For the details, see:
  - C. J. S. Alves and P. R. S. Antunes: "The Method of Fundamental Solutions applied to the calculation of eigenfrequencies and eigenmodes of 2D simply connected shapes," *Computers, Materials & Continua*, vol. 2, no. 4, pp. 251–266, 2005.
  - P. R. S. Antunes: "Numerical calculation of eigensolutions of 3D shapes using the Method of Fundamental Solutions," *Numerical Methods for Partial Differential Equations*, vol. 27, no. 6, pp. 1525–1550, 2011.

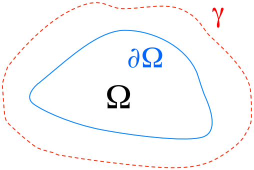
• The Method of Fundamental Solution (MFS)

<u>Fundamental solution:</u> $(\Delta + \lambda)\,\Phi_\lambda = -\delta$

$$\Phi_\lambda(x) = \begin{cases} \frac{i}{4} H_0^{(1)}(\sqrt{\lambda}|x|) & \text{in } \mathbb{R}^2 \\ \frac{e^{i\sqrt{\lambda}|x|}}{|x|} & \text{in } \mathbb{R}^3 \end{cases}$$

- Consider the approximation

$$u(x) \approx u_N(x) = \sum_{j=1}^{N} \alpha_j \Phi_\lambda(x - y_j)$$



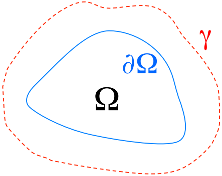
$y_j \in \gamma$

γ an admissible curve

- The coefficients are calculated such that $u_N(x)$ fits the boundary conditions

• Theoretical results

$$u(x) \approx u_N(x) = \sum_{j=1}^{N} \alpha_j \Phi_\lambda(x - y_j)$$



Given an open set $\Omega \subset \mathbb{R}^n$, $y_1, y_2, ..., y_N \in \bar{\Omega}^C$ different points and $\lambda \in \mathbb{R}$, then $\{\Phi_\lambda(x - y_1), ..., \Phi_\lambda(x - y_N)\}$ are linear independent on $\partial\Omega$.

If γ is the boundary of a domain which contains Ω, the set

$$Span\left(\{\Phi_\lambda(x - y)|_{x\in\Omega} : y \in \gamma\}\right)$$

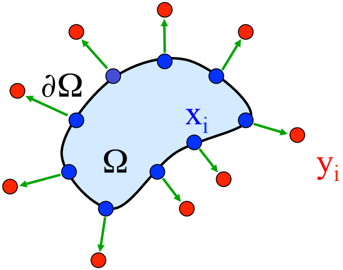
is dense in $H^1(\partial\Omega)$.

• Algorithm for the source points (2D)

$$u(x) \approx u_N(x) = \sum_{j=1}^{N} \alpha_j \Phi_\lambda(x - y_j)$$

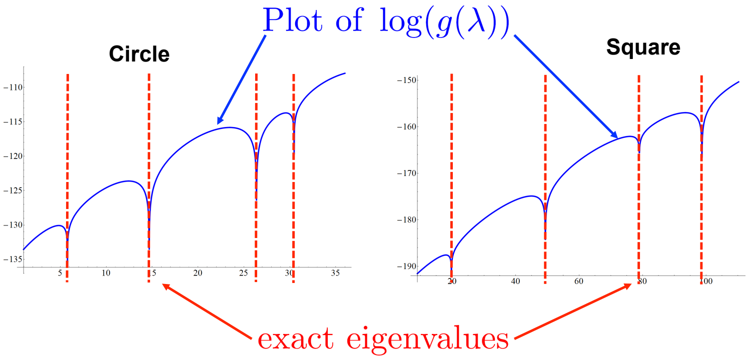
- Consider N points $x_1, \ldots, x_N \in \partial\Omega$ collocation points (*almost equally spaced*)
- Define N points $y_1, \ldots, y_N$ source points

$$y_i = x_i + \alpha\ n_i$$



• Algorithm for the eigenfrequency calculation

- Build the matrices $A_N(\lambda) = \Phi_\lambda(x_i - y_j)$
- Consider $g(\lambda) = |det(A_N(\lambda))|$ and look for the minima

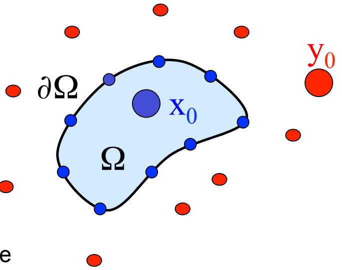


- Search for local minima using a direct search method

• Algorithm for the eigenfunction calculation

• Define extra points
$$\begin{cases} x_0 \in \Omega \\ y_0 \in \bar{\Omega}^C \end{cases}$$

The extra point x_0 is not on a nodal line



• Given the approximate eigenvalue λ, define

$$u(x) \approx \tilde{u}(x) = \sum_{j=0}^{N} \alpha_j \Phi_\lambda(x - y_j)$$

• To calculate α_i solve the system

$$\begin{cases} \tilde{u}(x_0) = 1 \\ \tilde{u}(x_i) = 0, \ i = 1, \ldots, N \end{cases}$$

- non null solution,
- null at boundary points

• Error bounds (Dirichlet case)

A posteriori bound (*Moler and Payne* 1968)

Let $(\tilde{\lambda}, \tilde{u})$ *be an approximation for the pair (eigenvalue,eigenfunction) which satisfies the problem*

$$\begin{cases} \Delta \tilde{u} + \tilde{\lambda}\tilde{u} = 0, \text{ in } \Omega \\ \tilde{u} = \xi(x), \text{ on } \partial\Omega \end{cases}$$

(with small ξ *)*

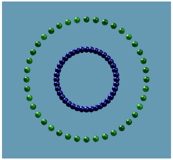
Then there exists an eigenvalue λ *and eigenfunction u such that*

$$\boxed{\frac{\tilde{\lambda}}{1+\theta} \le \lambda \le \frac{\tilde{\lambda}}{1-\theta}} \quad \text{and} \quad \boxed{\|u - \tilde{u}\|_{L^2(\Omega)} \le c_\Omega \theta}$$

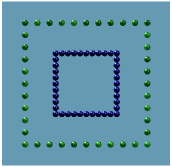
where $\theta = \dfrac{\sqrt{|\Omega|}\,\|\xi\|_{L^\infty(\partial\Omega)}}{\|\tilde{u}\|_{L^2(\Omega)}}$ *is very small if* $\tilde{u} \approx 0$ *on* $\partial\Omega$.

• Numerical tests (Dirichlet case) – 2D

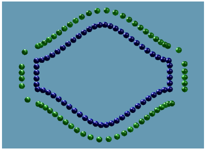
N=dimension of the matrix



N	abs. error (λ_1)	N	abs. error (λ_2)	N	abs. error (λ_3)
30	2.31×10^{-6}	30	4.94×10^{-6}	30	5.21×10^{-6}
40	5.91×10^{-8}	40	1.21×10^{-8}	40	1.26×10^{-7}
50	1.64×10^{-9}	50	3.01×10^{-10}	50	3.27×10^{-9}
60	8.23×10^{-11}	60	9.31×10^{-12}	60	9.35×10^{-11}

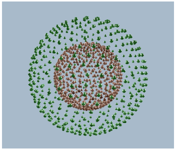


N	abs. error (λ_1)	N	abs. error (λ_2)	N	abs. error (λ_3)
30	5.72×10^{-6}	30	1.36×10^{-6}	30	1.81×10^{-5}
40	8.42×10^{-8}	40	1.67×10^{-7}	40	2.17×10^{-7}
50	7.76×10^{-8}	50	1.11×10^{-8}	50	6.94×10^{-8}
60	1.46×10^{-9}	60	1.44×10^{-9}	60	3.17×10^{-9}

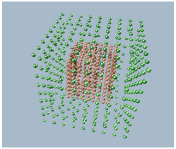


N	abs. error (λ_5)	N	abs. error (λ_5)	N	abs. error (λ_5)
20	2.11×10^{-4}	30	1.46×10^{-5}	40	1.23×10^{-6}
50	3.06×10^{-7}	60	2.52×10^{-8}	70	5.05×10^{-9}
80	3.19×10^{-9}	90	6.19×10^{-10}	100	1.87×10^{-10}

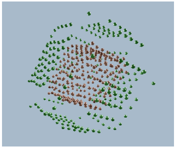
• Numerical tests (Dirichlet case) – 3D



N	abs. error (λ_1)	N	abs. error (λ_2)	N	abs. error (λ_3)
112	1.25×10^{-8}	112	9.21×10^{-7}	112	8.57×10^{-6}
158	8.61×10^{-12}	158	1.97×10^{-9}	158	6.53×10^{-8}
212	2.18×10^{-14}	212	1.61×10^{-13}	212	9.46×10^{-11}



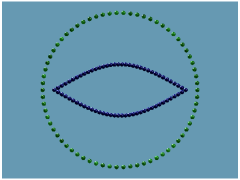
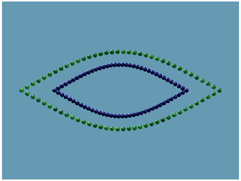
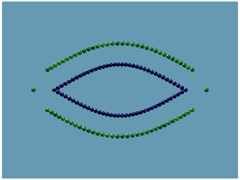
N	abs. error (λ_1)	N	abs. error (λ_2)	N	abs. error (λ_3)
218	6.13×10^{-10}	218	9.27×10^{-7}	218	1.55×10^{-6}
296	3.11×10^{-10}	296	7.31×10^{-8}	296	7.09×10^{-8}
386	9.15×10^{-12}	386	5.25×10^{-9}	386	1.95×10^{-10}



N	abs. error (λ_5)	N	abs. error (λ_5)	N	abs. error (λ_5)
226	1.36×10^{-5}	304	5.87×10^{-6}	374	7.21×10^{-8}

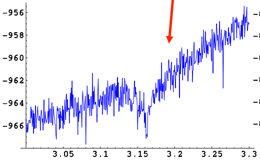
• Numerical tests (on the location of point sources)

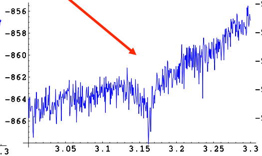
Point-sources on the boundary of a circular domain	Point-sources on an "expansion" of $\partial\Omega$	With the choice proposed

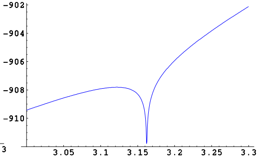




big rounding errors

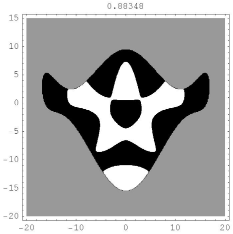
Plot of $\log(g(\lambda))$

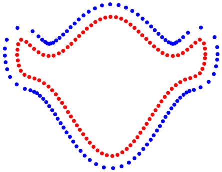


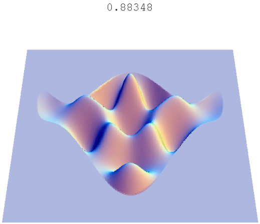




• Numerical Simulations

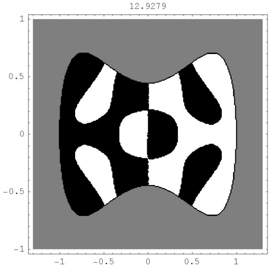


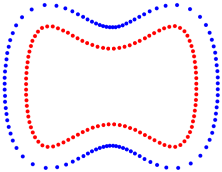


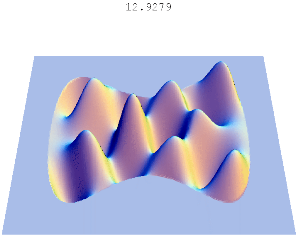


nodal domains plot

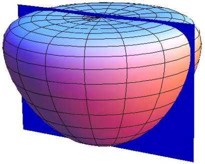
eigenfunction

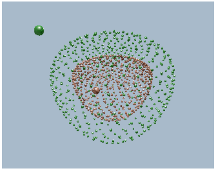




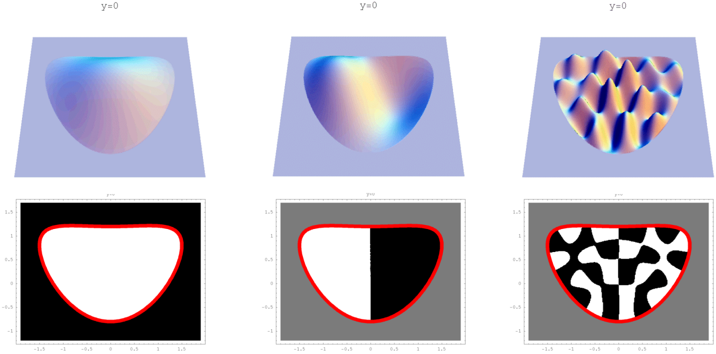


• Numerical simulations – non trivial domains 3D





3D plots of eigenfunctions associated to three eigenvalues



MFS Extensions

- The classical MFS is not accurate for corner/crack singularities
- However, splitting a solution into a regular part and a singular part combining MFS with the Method of Particular Solutions (Betcke/Trefethen), one can obtain highly accurate solutions.
- Reference: P. R. S. Antunes and S. S. Valtchev: "A meshfree numerical method for acoustic wave propagation problems in planar domains with corners and cracks," *J. Comput. Appl. Math.*, vol. 234, pp. 2646–2662, 2010.

MFS Extensions

- The classical MFS is not accurate for corner/crack singularities
- However, splitting a solution into a regular part and a singular part combining MFS with the Method of Particular Solutions (Betcke/Trefethen), one can obtain highly accurate solutions.
- Reference: P. R. S. Antunes and S. S. Valtchev: "A meshfree numerical method for acoustic wave propagation problems in planar domains with corners and cracks," *J. Comput. Appl. Math.*, vol. 234, pp. 2646–2662, 2010.

MFS Extensions

- The classical MFS is not accurate for corner/crack singularities
- However, splitting a solution into a regular part and a singular part combining MFS with the Method of Particular Solutions (Betcke/Trefethen), one can obtain highly accurate solutions.
- Reference: P. R. S. Antunes and S. S. Valtchev: "A meshfree numerical method for acoustic wave propagation problems in planar domains with corners and cracks," *J. Comput. Appl. Math.*, vol. 234, pp. 2646–2662, 2010.

A Potential Problem of MFS for Imaging/Inverse Problems

- MFS requires specific boundary condition to begin with (Dirichlet, Neumann, or Robin).
- In imaging and/or inverse problems, what is the natural boundary condition to use for a local region of interest (ROI)?
- The Dirichlet boundary condition $u|_{\partial\Omega} = 0$ is certainly not natural; the material value at the boundary shouldn't be 0.
- Furthermore, it may suffer from the *Gibbs phenomenon* (just like in truncated Fourier series).
- The Neumann boundary condition may be a bit better than the Dirichlet case: $\left.\frac{\partial u}{\partial \nu}\right|_{\partial\Omega} = 0$.
- Is it really natural to represent an ROI within a larger domain? Cannot expect the values (intensity) across the boundary $\partial\Omega$ are *flat*.

A Potential Problem of MFS for Imaging/Inverse Problems

- MFS requires specific boundary condition to begin with (Dirichlet, Neumann, or Robin).
- In imaging and/or inverse problems, what is the natural boundary condition to use for a local region of interest (ROI)?
- The Dirichlet boundary condition $u|_{\partial\Omega} = 0$ is certainly not natural; the material value at the boundary shouldn't be 0.
- Furthermore, it may suffer from the *Gibbs phenomenon* (just like in truncated Fourier series).
- The Neumann boundary condition may be a bit better than the Dirichlet case: $\left.\frac{\partial u}{\partial \nu}\right|_{\partial\Omega} = 0$.
- Is it really natural to represent an ROI within a larger domain? Cannot expect the values (intensity) across the boundary $\partial\Omega$ are *flat*.

A Potential Problem of MFS for Imaging/Inverse Problems

- MFS requires specific boundary condition to begin with (Dirichlet, Neumann, or Robin).
- In imaging and/or inverse problems, what is the natural boundary condition to use for a local region of interest (ROI)?
- The Dirichlet boundary condition $u|_{\partial\Omega} = 0$ is certainly not natural; the material value at the boundary shouldn't be 0.
- Furthermore, it may suffer from the *Gibbs phenomenon* (just like in truncated Fourier series).
- The Neumann boundary condition may be a bit better than the Dirichlet case: $\left.\frac{\partial u}{\partial \nu}\right|_{\partial\Omega} = 0$.
- Is it really natural to represent an ROI within a larger domain? Cannot expect the values (intensity) across the boundary $\partial\Omega$ are *flat*.

A Potential Problem of MFS for Imaging/Inverse Problems

- MFS requires specific boundary condition to begin with (Dirichlet, Neumann, or Robin).
- In imaging and/or inverse problems, what is the natural boundary condition to use for a local region of interest (ROI)?
- The Dirichlet boundary condition $u|_{\partial\Omega} = 0$ is certainly not natural; the material value at the boundary shouldn't be 0.
- Furthermore, it may suffer from the *Gibbs phenomenon* (just like in truncated Fourier series).
- The Neumann boundary condition may be a bit better than the Dirichlet case: $\left.\frac{\partial u}{\partial \nu}\right|_{\partial\Omega} = 0$.
- Is it really natural to represent an ROI within a larger domain? Cannot expect the values (intensity) across the boundary $\partial\Omega$ are *flat*.

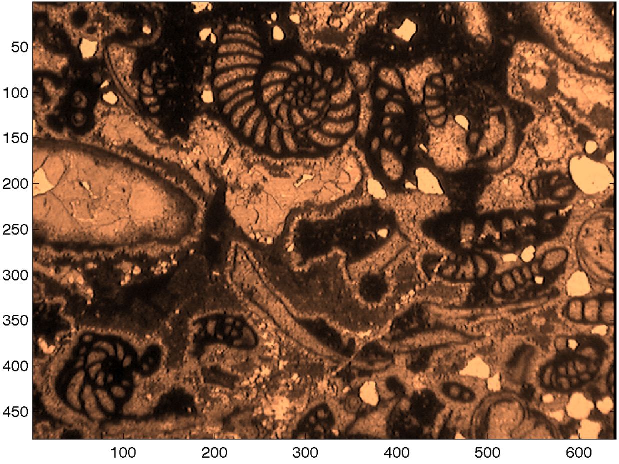
A Potential Problem of MFS for Imaging/Inverse Problems

- MFS requires specific boundary condition to begin with (Dirichlet, Neumann, or Robin).
- In imaging and/or inverse problems, what is the natural boundary condition to use for a local region of interest (ROI)?
- The Dirichlet boundary condition $u|_{\partial\Omega} = 0$ is certainly not natural; the material value at the boundary shouldn't be 0.
- Furthermore, it may suffer from the *Gibbs phenomenon* (just like in truncated Fourier series).
- The Neumann boundary condition may be a bit better than the Dirichlet case: $\left.\frac{\partial u}{\partial \nu}\right|_{\partial\Omega} = 0$.
- Is it really natural to represent an ROI within a larger domain? Cannot expect the values (intensity) across the boundary $\partial\Omega$ are *flat*.

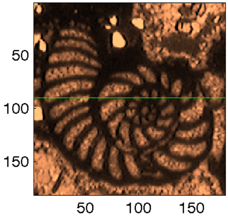
A Potential Problem of MFS for Imaging/Inverse Problems

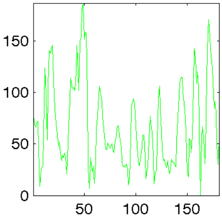
- MFS requires specific boundary condition to begin with (Dirichlet, Neumann, or Robin).
- In imaging and/or inverse problems, what is the natural boundary condition to use for a local region of interest (ROI)?
- The Dirichlet boundary condition $u|_{\partial\Omega} = 0$ is certainly not natural; the material value at the boundary shouldn't be 0.
- Furthermore, it may suffer from the *Gibbs phenomenon* (just like in truncated Fourier series).
- The Neumann boundary condition may be a bit better than the Dirichlet case: $\left.\frac{\partial u}{\partial \nu}\right|_{\partial\Omega} = 0$.
- Is it really natural to represent an ROI within a larger domain? Cannot expect the values (intensity) across the boundary $\partial\Omega$ are *flat*.

Photograph of Geological Specimen



Boundary Values of an ROI





Outline

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:

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

for all $\boldsymbol{x} \in \partial\Omega$, where φ is an eigenfunction common for both operators.

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:

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

for all $\boldsymbol{x} \in \partial\Omega$, where φ is an eigenfunction common for both operators.

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 $\mathcal{K}$ 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 $K(\boldsymbol{x},\boldsymbol{y})$ 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 $K(x, y)$ 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 $K(\boldsymbol{x}, \boldsymbol{y})$ 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 $\mathcal{K}$ 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 $\implies$ 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 $\tanh \frac{\sqrt{-\lambda_0}}{2} = \frac{2}{\sqrt{-\lambda_0}}$,

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

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

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

- λ_{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 $\tanh \frac{\sqrt{-\lambda_0}}{2} = \frac{2}{\sqrt{-\lambda_0}}$,

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

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

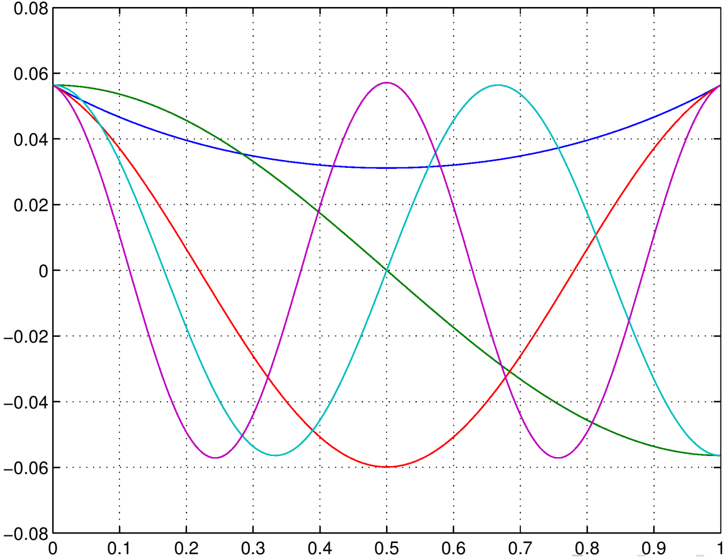
$$\varphi_{2m-1}(x) = \sqrt{2}\cos(2m-1)\pi x;$$

- $\lambda_{2m},\ m = 1, 2, \ldots,$ 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.

1D Example: Rayleigh Functions/Trace Formula

Corollary (NS 2008)

Let $\{\lambda_n\}_{n=0}^{\infty}$ be the 1D Laplacian eigenvalues of the non-local boundary problem with the commuting integral operator whose kernel is $K(x,y) = -|x-y|/2$. Then, they satisfy the following trace formula:

$$\sum_{n=0}^{\infty} \frac{1}{\lambda_n} = \int_0^1 K(x,x)\,\mathrm{d}x = 0.$$

Compare this with the famous Basel problem, which is based on the Dirichlet boundary condition:

$$\sum_{n=1}^{\infty} \frac{1}{\pi^2 n^2} = \int_0^1 G_D(x,x)\,\mathrm{d}x = \frac{1}{6} \iff \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}.$$

1D Example: Rayleigh Functions/Trace Formula

Corollary (NS 2008)

Let $\{\lambda_n\}_{n=0}^{\infty}$ *be the 1D Laplacian eigenvalues of the non-local boundary problem with the commuting integral operator whose kernel is* $K(x,y) = -|x-y|/2$. *Then, they satisfy the following trace formula:*

$$\sum_{n=0}^{\infty} \frac{1}{\lambda_n} = \int_0^1 K(x,x)\,\mathrm{d}x = 0.$$

Compare this with the famous Basel problem, which is based on the Dirichlet boundary condition:

$$\sum_{n=1}^{\infty} \frac{1}{\pi^2 n^2} = \int_0^1 G_D(x,x)\,\mathrm{d}x = \frac{1}{6} \quad \Longleftrightarrow \quad \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}.$$

1D Example: Rayleigh Functions/Trace Formula ...

Theorem (NS 2008)

Let $K_p(x,y)$ be the pth iterated kernel of $K(x,y) = -|x-y|/2$. Then,

$$\sum_{n=0}^{\infty} \frac{1}{\lambda_n^p} = \int_0^1 K_p(x,x)\,\mathrm{d}x = \frac{1}{4^p}\left(S_{2p} + \frac{(-1)^p}{\alpha^{2p}}\right) + \frac{4^p-1}{2\cdot(2p)!}|B_{2p}|,$$

where $\alpha \approx 1.19967864$ satisfies $\alpha = \coth\alpha$, B_{2p} is the Bernoulli number, and

$$S_{2p} := \sum_{m=1}^{\infty}\left(\frac{4}{\lambda_{2m}}\right)^p,$$

satisfies the following recursion formula:

$$\sum_{\ell=1}^{n+1} \frac{(-1)^{n-\ell+1}(2(n-\ell+1)-1)}{(2(n-\ell+1))!}\left\{S_{2\ell} + \frac{(-1)^\ell}{\alpha^{2\ell}}\right\} = \frac{(-1)^n}{2(2n)!}.$$

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

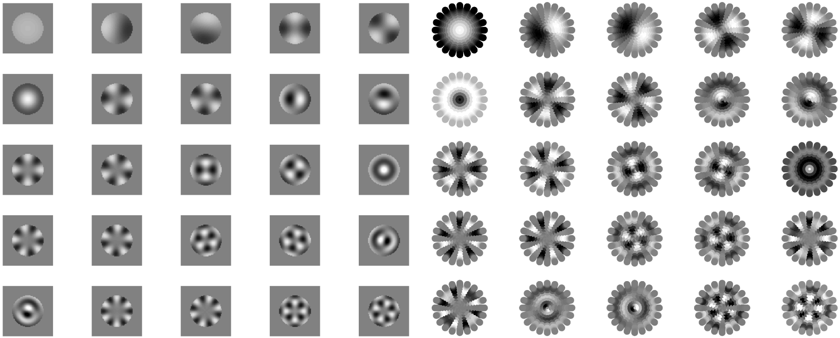
$$\varphi_{m,n}(r,\theta) = \begin{cases} J_m(\beta_{m-1,n}\,r)\binom{\cos}{\sin}(m\theta) & \text{if } m=1,2,\ldots,\ n=1,2,\ldots, \\ J_0(\beta_{0,n}\,r) & \text{if } m=0,\ n=1,2,\ldots, \end{cases}$$

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

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 $\beta_{k,\ell}$ is the ℓth zero of the Bessel function of order k, $J_k(\beta_{k,\ell})=0$. Then,
$$\varphi_{m,n}(r,\theta) = \begin{cases} J_m(\beta_{m-1,n}\,r)\binom{\cos}{\sin}(m\theta) & \text{if } m=1,2,\ldots,\ n=1,2,\ldots, \\ J_0(\beta_{0,n}\,r) & \text{if } m=0,\ n=1,2,\ldots, \end{cases}$$
$$\lambda_{m,n} = \begin{cases} \beta^2_{m-1,n}, & \text{if } m=1,\ldots,\ n=1,2,\ldots, \\ \beta^2_{0,n} & \text{if } m=0,\ n=1,2,\ldots. \end{cases}$$

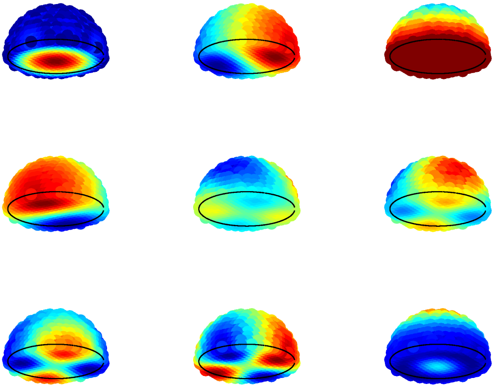
First 25 Basis Functions



(a) Our Basis (b) Dirichlet-Laplace

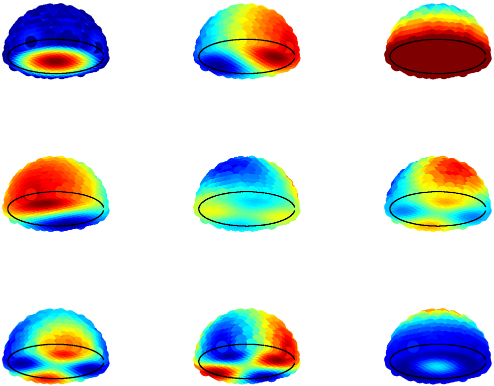
3D Example

- 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

Connection with Potential Theory

- Mark Kac mentioned at the very end of his 1951 paper (Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability) that the same integral equation in 3D is equivalent to the Laplacian eigenvalue problem. But his BC was incorrect.
- In 1967–9, John Troutman studied the eigenvalues of the same integral operator (i.e., the logarithmic potential) in 2D. He posed this problem as the Laplacian eigenvalue problem whose eigenfunctions are harmonic outside of the given domain. He proved that there exists one negative eigenvalue iff the *transfinite diameter* (or *logarithmic capacity*) of the closed domain $\overline{\Omega}$ exceeds 1.
- In 1970, Mark Kac and Tomasz Bojdecki obtained similar results using probabilistic argument (Kac) and purely analytic method (Bojdecki).

Connection with Potential Theory

- Mark Kac mentioned at the very end of his 1951 paper (Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability) that the same integral equation in 3D is equivalent to the Laplacian eigenvalue problem. But his BC was incorrect.
- In 1967–9, John Troutman studied the eigenvalues of the same integral operator (i.e., the logarithmic potential) in 2D. He posed this problem as the Laplacian eigenvalue problem whose eigenfunctions are harmonic outside of the given domain. He proved that there exists one negative eigenvalue iff the *transfinite diameter* (or *logarithmic capacity*) of the closed domain $\overline{\Omega}$ exceeds 1.
- In 1970, Mark Kac and Tomasz Bojdecki obtained similar results using probabilistic argument (Kac) and purely analytic method (Bojdecki).

Connection with Potential Theory

- Mark Kac mentioned at the very end of his 1951 paper (Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability) that the same integral equation in 3D is equivalent to the Laplacian eigenvalue problem. But his BC was incorrect.
- In 1967–9, John Troutman studied the eigenvalues of the same integral operator (i.e., the logarithmic potential) in 2D. He posed this problem as the Laplacian eigenvalue problem whose eigenfunctions are harmonic outside of the given domain. He proved that there exists one negative eigenvalue iff the *transfinite diameter* (or *logarithmic capacity*) of the closed domain $\overline{\Omega}$ exceeds 1.
- In 1970, Mark Kac and Tomasz Bojdecki obtained similar results using probabilistic argument (Kac) and purely analytic method (Bojdecki).

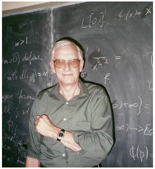
Connection with Potential Theory . . .

Since then, there have been some sporadic related works, but the use of the eigenfunctions of such potential operators has not been systematically pursued as far as we know.

Connection with Potential Theory . . .

Since then, there have been some sporadic related works, but the use of the eigenfunctions of such potential operators has not been systematically pursued as far as we know.

(a) Mark Kac (1914–1984)



(b) John Troutman (193?–)

(c) Tomasz Bojdecki (?)

Connection with Volterra Operators

- The 1959 paper of Victor B. Lidskiĭ "Conditions for completeness of a system of root subspaces for non-selfadjoint operators with discrete spectra," *Amer. Math. Soc. Transl. Ser. 2*, vol. 34, pp. 241–281, 1963, discusses the *iterated Volterra integral operator*:

$$Af(x) := \int_x^1 f(y)\,\mathrm{d}y,\ f \in L^2(0,1) \Longrightarrow A^2 f(x) = \int_x^1 (x-y) f(y)\,\mathrm{d}y$$

which was decomposed into the real and imaginary parts:

$$\begin{aligned}(A^2)_R f &:= \frac{1}{2}(A^2 + A^{2*}) = -\frac{1}{2}\int_0^1 |x-y| f(y)\,\mathrm{d}y;\\ (A^2)_I f &:= \frac{1}{2\mathrm{i}}(A^2 - A^{2*}) = \frac{1}{2\mathrm{i}}\int_0^1 (x-y) f(y)\,\mathrm{d}y.\end{aligned}$$

Connection with Volterra Operators . . .

- The famous book of Gohberg-Kreĭn (*Introduction to the Theory of Linear Nonselfadjoint Operators*, AMS, 1969) also discusses the same operators.
- Do the higher dimensional cases have also similar correspondence?

Connection with Volterra Operators . . .

- The famous book of Gohberg-Kreĭn (*Introduction to the Theory of Linear Nonselfadjoint Operators*, AMS, 1969) also discusses the same operators.
- Do the higher dimensional cases have also similar correspondence?

Connection with Volterra Operators . . .

- The famous book of Gohberg-Kreĭn (*Introduction to the Theory of Linear Nonselfadjoint Operators*, AMS, 1969) also discusses the same operators.
- Do the higher dimensional cases have also similar correspondence?

(a) Victor Lidskiĭ (1924–2008)

(b) Mark Krein (1907–1989)

(c) Israel Gohberg (1928–2009)

Connection with von Neumann–Kreĭn Extension Theory

- John von Neumann (1929) and Mark Kreĭn (1947) considered a *self-adjoint extension of symmetric operators*.
- Let $T := -\frac{\mathrm{d}^2}{\mathrm{d}x^2}$, $\mathscr{D}(T) := H_0^2(0,1) \subset H^2(0,1)$, where $H_0^2(0,1) := \left\{ f \in H^2(0,1) \mid f(0) = f(1) = f'(0) = f'(1) = 0 \right\}$ and $H^2(0,1) := \left\{ f \in C^1[0,1] \mid f' \in AC[0,1], f'' \in L^2(0,1) \right\}$. T is a positive *symmetric* operator on $\mathscr{D}(T)$, but *not self-adjoint* because $\mathscr{D}(T^*) = H^2(0,1) \supsetneqq \mathscr{D}(T)$.
- von Neumann-Kreĭn extension of T is the smallest (or soft) self-adjoint extension $T_0 = -\frac{\mathrm{d}^2}{\mathrm{d}x^2}$, $\mathscr{D}(T_0) = \left\{ f \in H^2(0,1) \mid f'(0) = f'(1) = f(1) - f(0) \right\} = \mathscr{D}(T_0^*)$.

Connection with von Neumann–Kreĭn Extension Theory

- John von Neumann (1929) and Mark Kreĭn (1947) considered a *self-adjoint extension of symmetric operators*.
- Let $T := -\frac{\mathrm{d}^2}{\mathrm{d}x^2}$, $\mathscr{D}(T) := H_0^2(0,1) \subset H^2(0,1)$, where $H_0^2(0,1) := \{f \in H^2(0,1) \mid f(0) = f(1) = f'(0) = f'(1) = 0\}$ and $H^2(0,1) := \{f \in C^1[0,1] \mid f' \in AC[0,1], f'' \in L^2(0,1)\}$. T is a positive *symmetric* operator on $\mathscr{D}(T)$, but *not self-adjoint* because $\mathscr{D}(T^*) = H^2(0,1) \supsetneq \mathscr{D}(T)$.
- von Neumann-Kreĭn extension of T is the smallest (or soft) self-adjoint extension $T_0 = -\frac{\mathrm{d}^2}{\mathrm{d}x^2}$, $\mathscr{D}(T_0) = \{f \in H^2(0,1) \mid f'(0) = f'(1) = f(1) - f(0)\} = \mathscr{D}(T_0^*)$.

Connection with von Neumann–Kreĭn Extension Theory

- John von Neumann (1929) and Mark Kreĭn (1947) considered a *self-adjoint extension of symmetric operators*.
- Let $T := -\frac{\mathrm{d}^2}{\mathrm{d}x^2}$, $\mathscr{D}(T) := H_0^2(0,1) \subset H^2(0,1)$, where $H_0^2(0,1) := \{f \in H^2(0,1) \mid f(0) = f(1) = f'(0) = f'(1) = 0\}$ and $H^2(0,1) := \{f \in C^1[0,1] \mid f' \in AC[0,1], f'' \in L^2(0,1)\}$. T is a positive *symmetric* operator on $\mathscr{D}(T)$, but *not self-adjoint* because $\mathscr{D}(T^*) = H^2(0,1) \supsetneq \mathscr{D}(T)$.
- von Neumann-Kreĭn extension of T is the smallest (or soft) self-adjoint extension $T_0 = -\frac{\mathrm{d}^2}{\mathrm{d}x^2}$, $\mathscr{D}(T_0) = \{f \in H^2(0,1) \mid f'(0) = f'(1) = f(1) - f(0)\} = \mathscr{D}(T_0^*)$.

Connection with von Neumann–Kreĭn Extension Theory . . .

- Compare it with our boundary condition: $-f'(0) = f'(1) = f(0) + f(1)$.
- Also, compare it with the *Friedrichs extension* of T, which is the *largest (or hard) self-adjoint extension*: $T_\infty = -\frac{d^2}{dx^2}$, $\mathscr{D}(T_\infty) = \left\{ f \in H^2(0,1) \mid f(0) = f(1) = 0 \right\} = \mathscr{D}(T_\infty^*) \Longleftrightarrow$ Dirichlet BC!

Connection with von Neumann–Kreĭn Extension Theory . . .

- Compare it with our boundary condition: $-f'(0) = f'(1) = f(0) + f(1)$.
- Also, compare it with the *Friedrichs extension* of T, which is the *largest (or hard) self-adjoint extension*: $T_\infty = -\frac{\mathrm{d}^2}{\mathrm{d}x^2}$, $\mathscr{D}(T_\infty) = \left\{ f \in H^2(0,1) \mid f(0) = f(1) = 0 \right\} = \mathscr{D}(T_\infty^*) \Longleftrightarrow$ Dirichlet BC!

Connection with von Neumann–Kreĭn Extension Theory . . .

- Compare it with our boundary condition: $-f'(0) = f'(1) = f(0) + f(1)$.
- Also, compare it with the *Friedrichs extension* of T, which is the *largest (or hard) self-adjoint extension*: $T_\infty = -\frac{d^2}{dx^2}$, $\mathscr{D}(T_\infty) = \{f \in H^2(0,1) \mid f(0) = f(1) = 0\} = \mathscr{D}(T_\infty^*) \Longleftrightarrow$ Dirichlet BC!

(a) John von Neumann (1903–1957)

(b) Mark Krein (1907–1989)

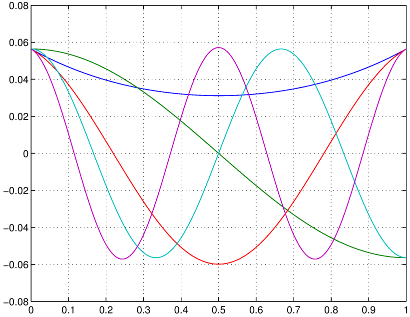
(c) Kurt Friedrichs (1901–1982)

Connection with von Neumann–Kreĭn Extension Theory . . .

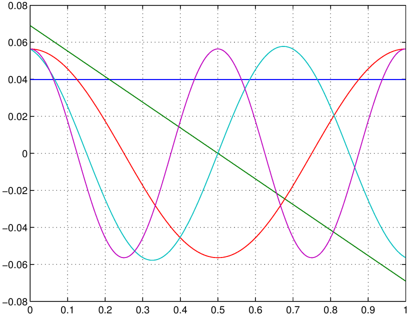
	Our Basis	Kreĭn-Laplacian Basis
λ_0	-5.756915. . . ; $\tanh\sqrt{-\lambda_0}/2 = 2/\sqrt{-\lambda_0}$	0
φ_0	$\cosh\sqrt{-\lambda_0}(x-1/2)$	1
λ_{2m-1}	$((2m-1)\pi)^2$	$\tan\sqrt{\lambda_{2m-1}}/2 = \sqrt{\lambda_{2m-1}}/2$
φ_{2m-1}	$\sin(2m-1)\pi(x-1/2)$	$\sin\sqrt{\lambda_{2m-1}}(x-1/2)$
λ_{2m}	$\tan\sqrt{\lambda_{2m}}/2 = -2/\sqrt{\lambda_{2m}}$	$(2m\pi)^2$
φ_{2m}	$\cos\sqrt{\lambda_{2m}}(x-1/2)$	$\cos 2m\pi(x-1/2)$

Note that the above eigenfunctions are not normalized to have $\|\cdot\|_2 = 1$.

Connection with von Neumann–Kreĭn Extension Theory . . .



(a) Our Basis



(b) Kreĭn-Laplacian Basis

Connection with von Neumann–Kreĭn Extension Theory . . .

- In higher dimensions, the von Neumann-Kreĭn extension of the Laplacian $T = -\Delta$, $\mathscr{D}(T) = H_0^2(\Omega)$, on $\Omega \subset \mathbb{R}^d$ turns out to be: $T_0 = -\Delta$, $\mathscr{D}(T_0) = \left\{ f \in H^2(\Omega) \,\middle|\, \frac{\partial f}{\partial \nu}(\boldsymbol{x}) = \frac{\partial H(f)}{\partial \nu}(\boldsymbol{x}), \boldsymbol{x} \in \partial\Omega \right\}$ where $H(f)$ is a harmonic function in Ω with the boundary condition: $H(f) = f$ on $\partial\Omega$; See e.g., A. Alonso & B. Simon: "The Birman-Kreĭn-Vishik theory of self-adjoint extensions of semibounded operators," *J. Operator Theory*, vol. 4, pp. 251–270, 1980.
- This is closely related to our Polyharmonic Local Sine Transform (PHLST): N. Saito & J.-F. Remy: "The polyharmonic local sine transform: A new tool for local image analysis and synthesis without edge effect," *Appl. Comput. Harm. Anal.*, vol. 20, pp. 41–73, 2006.
- After all, the von Neumann-Kreĭn extensions do not deal with the exterior of the domain Ω while our approach based on the commuting integral operators allow us to extend our eigenfunctions very naturally to the exterior of Ω.

Connection with von Neumann–Kreĭn Extension Theory . . .

- In higher dimensions, the von Neumann-Kreĭn extension of the Laplacian $T = -\Delta$, $\mathscr{D}(T) = H_0^2(\Omega)$, on $\Omega \subset \mathbb{R}^d$ turns out to be: $T_0 = -\Delta$, $\mathscr{D}(T_0) = \left\{ f \in H^2(\Omega) \,\middle|\, \frac{\partial f}{\partial \nu}(\boldsymbol{x}) = \frac{\partial H(f)}{\partial \nu}(\boldsymbol{x}), \boldsymbol{x} \in \partial\Omega \right\}$ where $H(f)$ is a harmonic function in Ω with the boundary condition: $H(f) = f$ on $\partial\Omega$; See e.g., A. Alonso & B. Simon: "The Birman-Kreĭn-Vishik theory of self-adjoint extensions of semibounded operators," *J. Operator Theory*, vol. 4, pp. 251–270, 1980.
- This is closely related to our Polyharmonic Local Sine Transform (PHLST): N. Saito & J.-F. Remy: "The polyharmonic local sine transform: A new tool for local image analysis and synthesis without edge effect," *Appl. Comput. Harm. Anal.*, vol. 20, pp. 41–73, 2006.
- After all, the von Neumann-Kreĭn extensions do not deal with the exterior of the domain Ω while our approach based on the commuting integral operators allow us to extend our eigenfunctions very naturally to the exterior of Ω.

Connection with von Neumann–Kreĭn Extension Theory . . .

- In higher dimensions, the von Neumann-Kreĭn extension of the Laplacian $T = -\Delta$, $\mathscr{D}(T) = H_0^2(\Omega)$, on $\Omega \subset \mathbb{R}^d$ turns out to be: $T_0 = -\Delta$, $\mathscr{D}(T_0) = \left\{ f \in H^2(\Omega) \,\middle|\, \frac{\partial f}{\partial \nu}(\boldsymbol{x}) = \frac{\partial H(f)}{\partial \nu}(\boldsymbol{x}), \boldsymbol{x} \in \partial\Omega \right\}$ where $H(f)$ is a harmonic function in Ω with the boundary condition: $H(f) = f$ on $\partial\Omega$; See e.g., A. Alonso & B. Simon: "The Birman-Kreĭn-Vishik theory of self-adjoint extensions of semibounded operators," *J. Operator Theory*, vol. 4, pp. 251–270, 1980.
- This is closely related to our Polyharmonic Local Sine Transform (PHLST): N. Saito & J.-F. Remy: "The polyharmonic local sine transform: A new tool for local image analysis and synthesis without edge effect," *Appl. Comput. Harm. Anal.*, vol. 20, pp. 41–73, 2006.
- After all, the von Neumann-Kreĭn extensions do not deal with the exterior of the domain Ω while our approach based on the commuting integral operators allow us to extend our eigenfunctions very naturally to the exterior of Ω.

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.

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

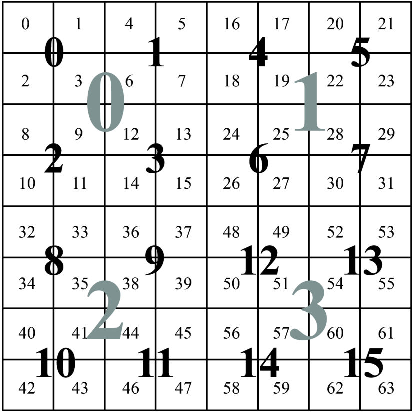
- Let $K_{i,j} := w_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\varphi_i := \varphi(\boldsymbol{x}_i)$, and $\boldsymbol{\varphi} := (\varphi_1, \ldots, \varphi_N)^{\mathsf{T}} \in \mathbb{R}^N$. Then, the above equation can be written in a matrix-vector format as: $K\boldsymbol{\varphi} = \mu\boldsymbol{\varphi}$, 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

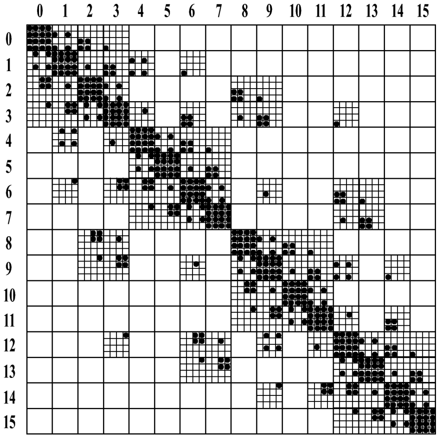
A Possible Fast Algorithm for Computing φ_j's

- 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

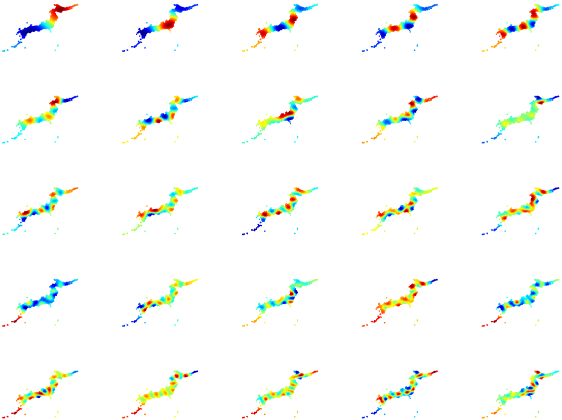


(a) Hierarchical indexing scheme

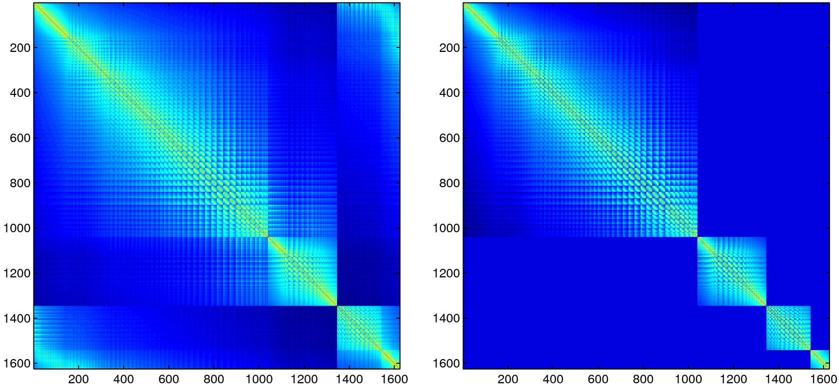


(b) Tree-Structured Matrix

First 25 Basis Functions via the FMM-based algorithm



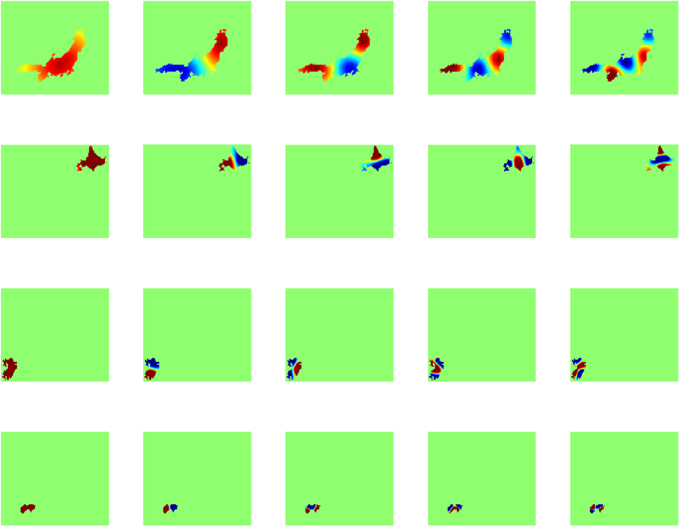
Splitting into Subproblems for Faster Computation



(a) Whole islands

(b) Separated islands

Eigenfunctions for Separated Islands



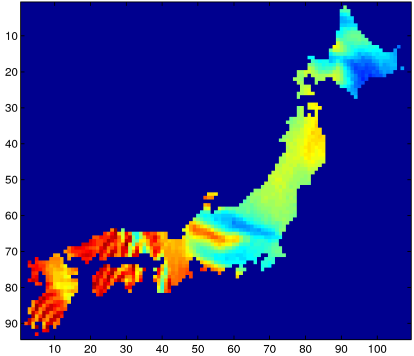
Outline

Applications

- Suppose images (or vector-valued measurements) are recorded on the domain Ω of general shape in $\mathbb{R}^d$; $d = 2$ or 3.
- 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.

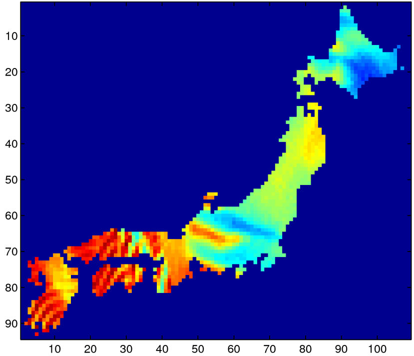
Outline

Image Approximation; Comparison with Wavelets

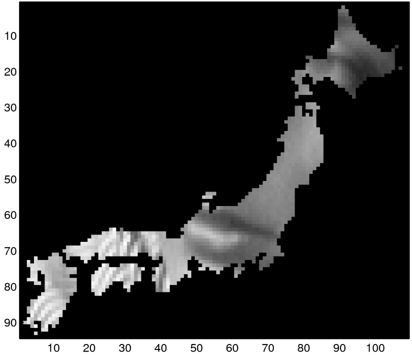


(a) What data?

Image Approximation; Comparison with Wavelets

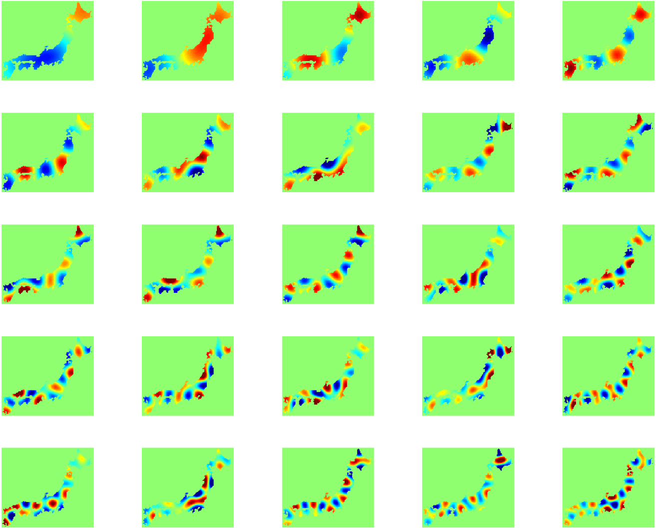


(a) What data?

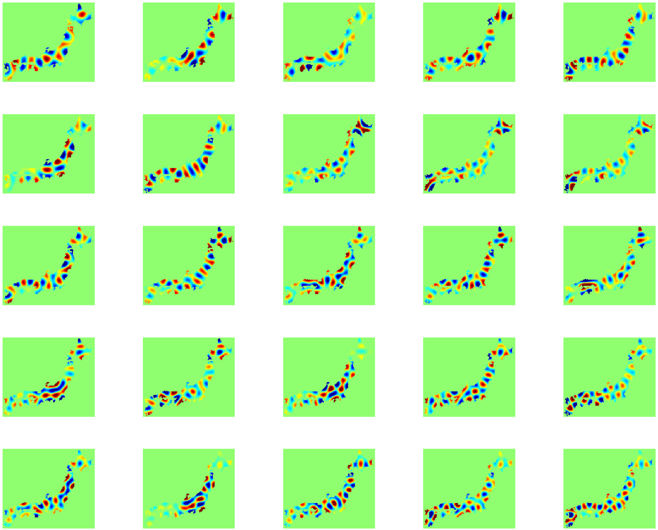


(b) $\chi_J \cdot$ Barbara

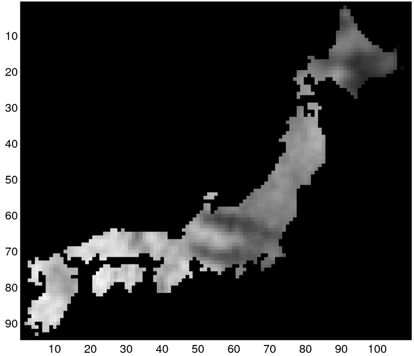
First 25 Basis Functions



Next 25 Basis Functions

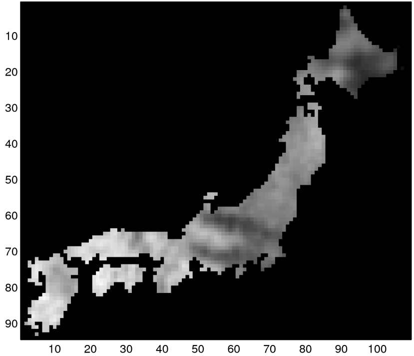


Reconstruction with Top 100 Coefficients

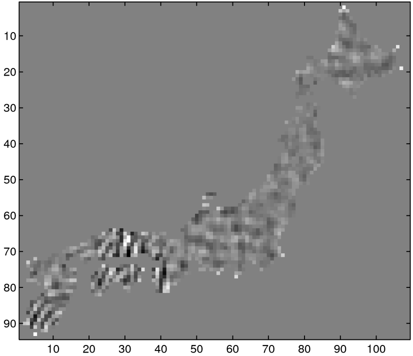


(a) Reconstruction

Reconstruction with Top 100 Coefficients

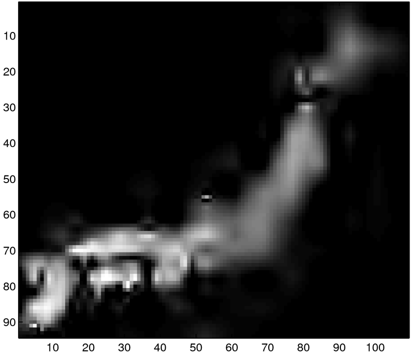


(a) Reconstruction



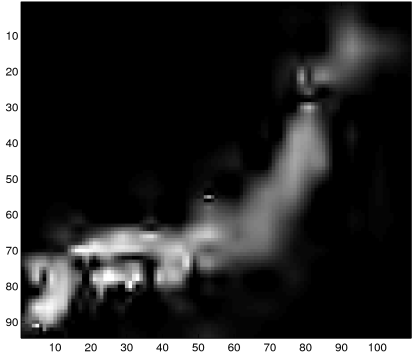
(b) Error

Reconstruction with Top 100 2D Wavelets (Symmlet 8)

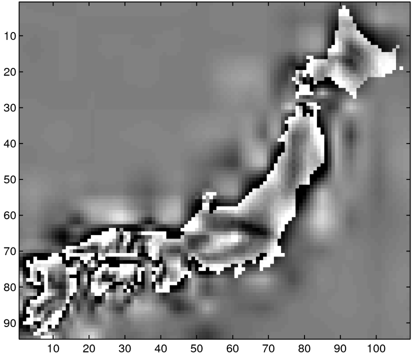


(a) Reconstruction

Reconstruction with Top 100 2D Wavelets (Symmlet 8)

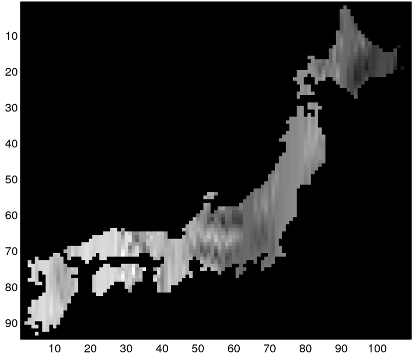


(a) Reconstruction



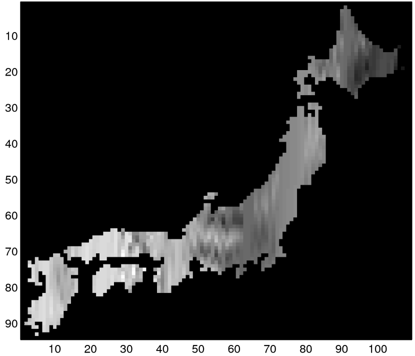
(b) Error

Reconstruction with Top 100 1D Wavelets (Symmlet 8)

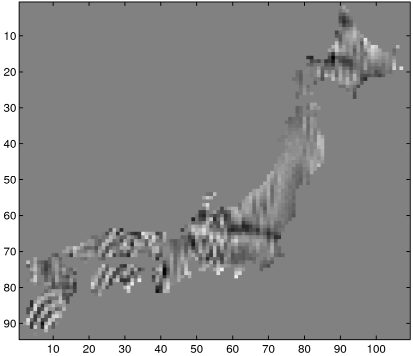


(a) Reconstruction

Reconstruction with Top 100 1D Wavelets (Symmlet 8)

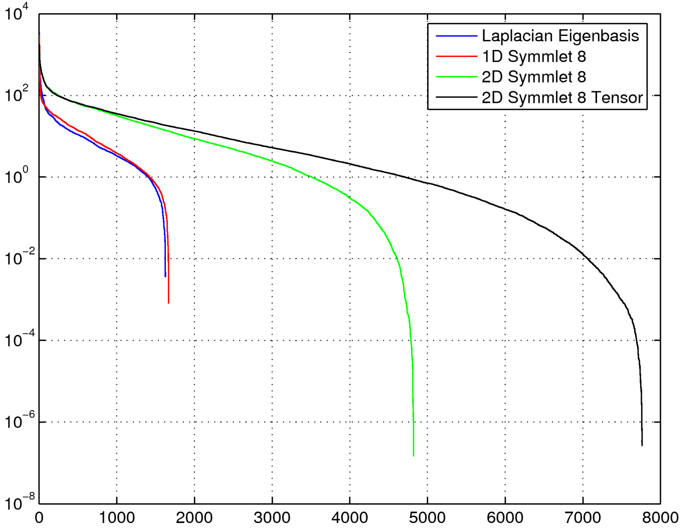


(a) Reconstruction

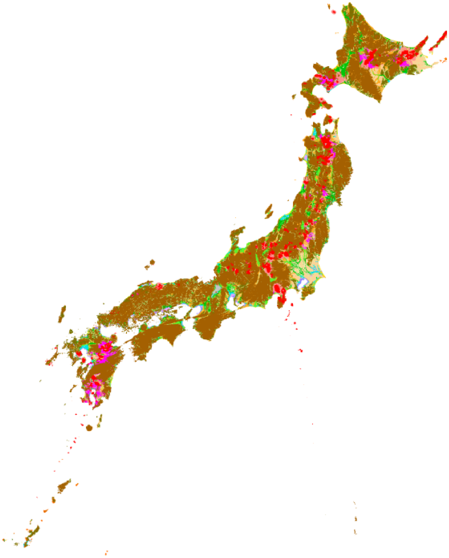


(b) Error

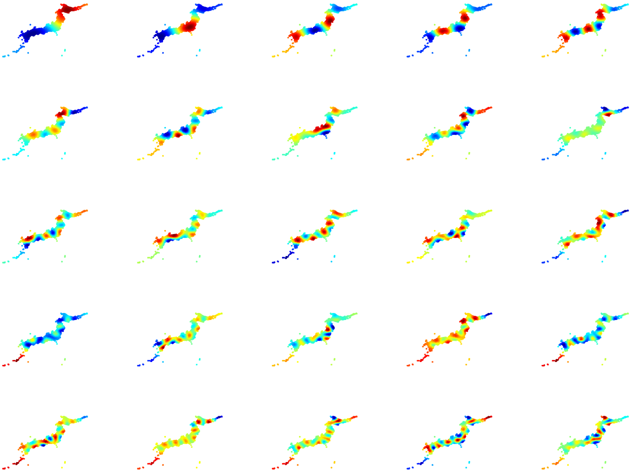
Comparison of Coefficient Decay



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



First 25 Basis Functions via the FMM-based algorithm



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;

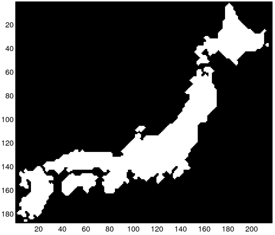
Ω_3: The same as Ω_2 but with a "jaggy" boundary curve

Ω_4: The two-component version of Ω_2.

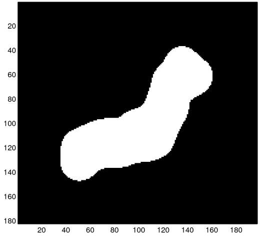
As for the data on these domains, we adopted three functions with different smoothness:

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.

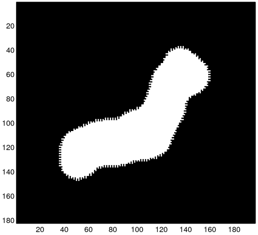
The Domains with Perturbed Boundaries



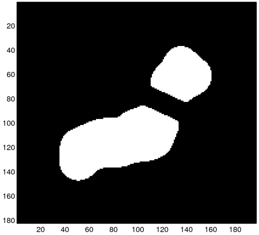
(a) χ_{Ω_1}



(b) χ_{Ω_2}

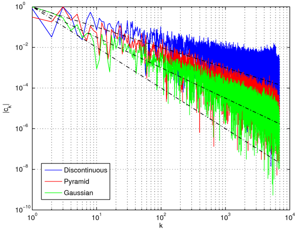


(c) χ_{Ω_3}

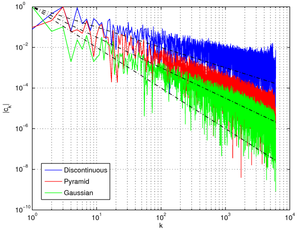


(d) χ_{Ω_4}

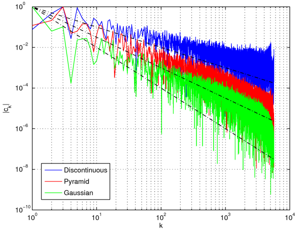
Decay Rates of the Expansion Coefficients (Unsorted)



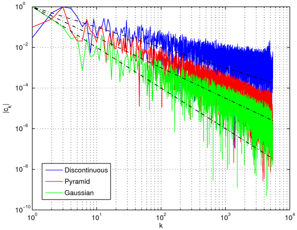
(a) Decay rates on Ω_1



(b) Decay rates on Ω_2



(c) Decay rates on Ω_3

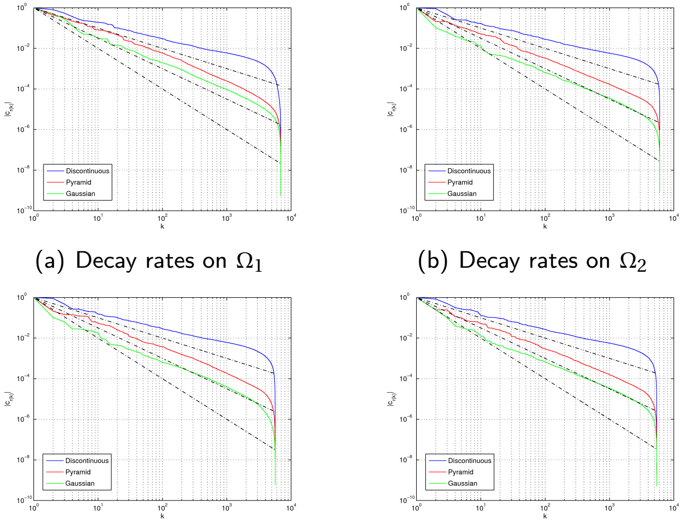


(d) Decay rates on Ω_4

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

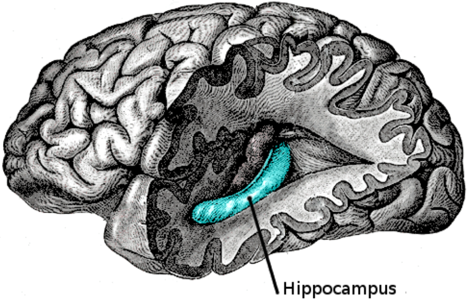
Conjecture (NS 2007)

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* $m \to \infty$*.*

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



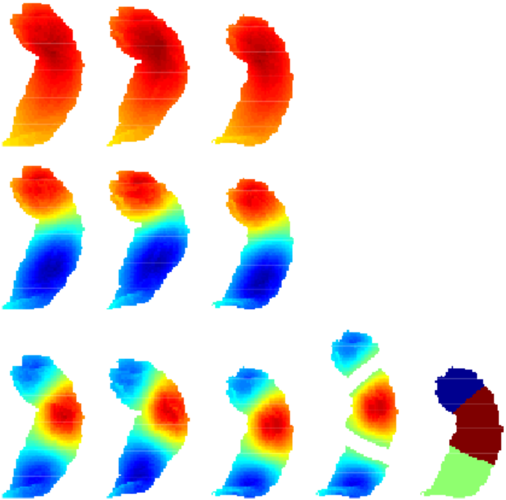
Hippocampal Shape Analysis ...

- 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:

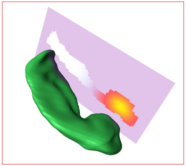
$$\boldsymbol{F} := \left(\frac{\lambda_1}{\lambda_2}, \ldots, \frac{\lambda_1}{\lambda_{n+1}}\right)^{\mathsf{T}} = \left(\frac{\mu_2}{\mu_1}, \ldots, \frac{\mu_{n+1}}{\mu_1}\right)^{\mathsf{T}} \in \mathbb{R}^n.$$

  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)

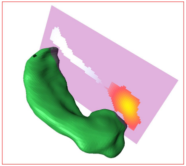
First Three Eigenfunctions of Three Patients



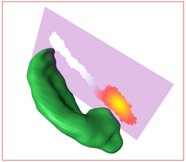
The Second Eigenfunction φ_2



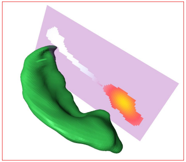
(a) $N = 15135$



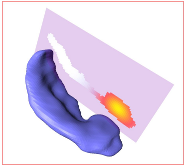
(b) $N = 15438$



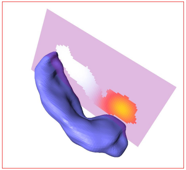
(c) $N = 14938$



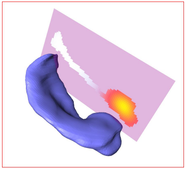
(d) $N = 15256$



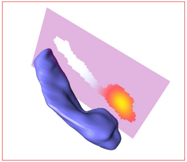
(e) $N = 14201$



(f) $N = 15630$

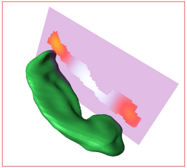


(g) $N = 12073$

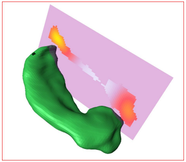


(h) $N = 12240$

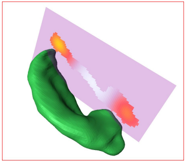
The Third Eigenfunction φ_3



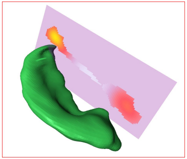
(i) $N = 15135$



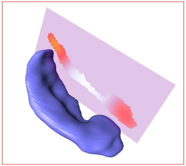
(j) $N = 15438$



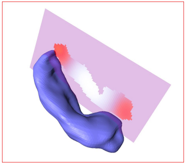
(k) $N = 14938$



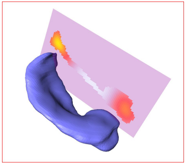
(l) $N = 15256$



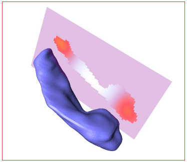
(m) $N = 14201$



(n) $N = 15630$



(o) $N = 12073$



(p) $N = 12240$

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

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

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

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

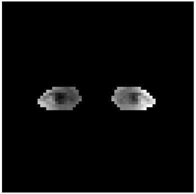
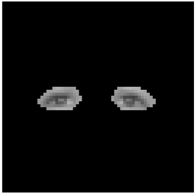
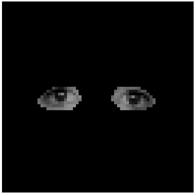
Outline

Comparison with PCA

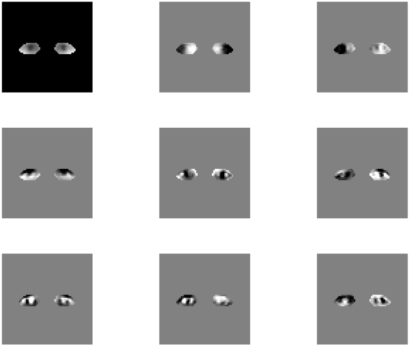
- Consider a stochastic process living on a domain Ω.
- *PCA/Karhunen-Loève Transform* is often used.
- PCA/KLT *implicitly* incorporate geometric information of the measurement (or pixel) location through *data correlation*.
- Our Laplacian eigenfunctions use *explicit* geometric information through 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

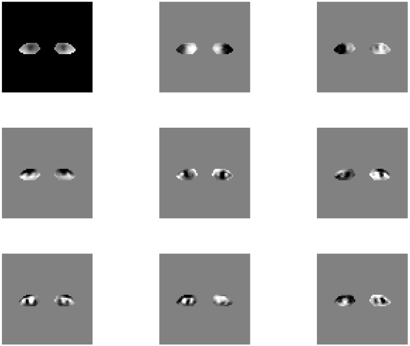




Comparison with PCA: Basis Vectors

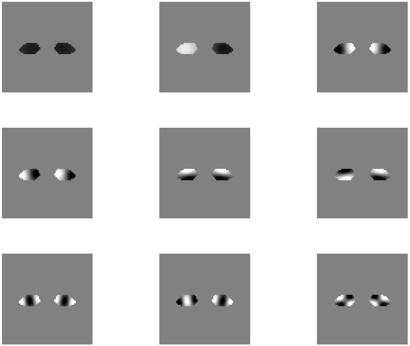


(a) KLB/PCA 1:9

Comparison with PCA: Basis Vectors

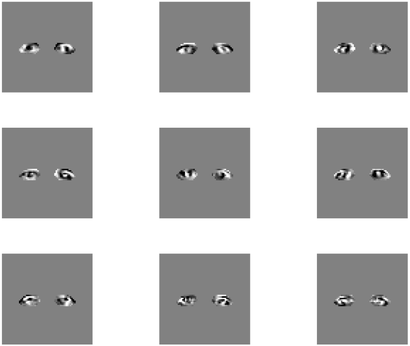


(a) KLB/PCA 1:9

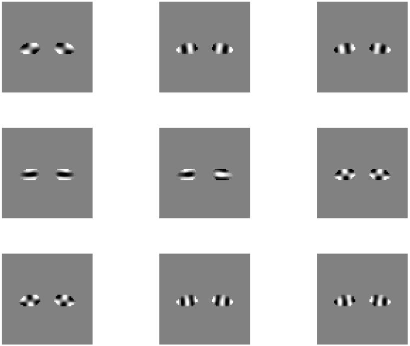


(b) Laplacian Eigenfunctions 1:9

Comparison with PCA: Basis Vectors . . .

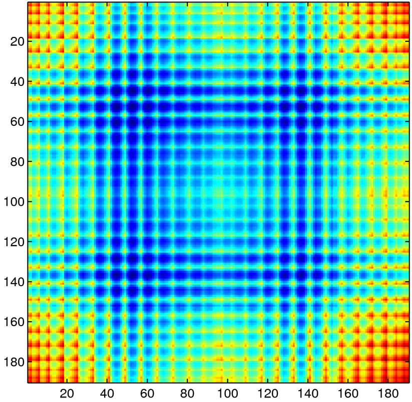


(a) KLB/PCA 10:18

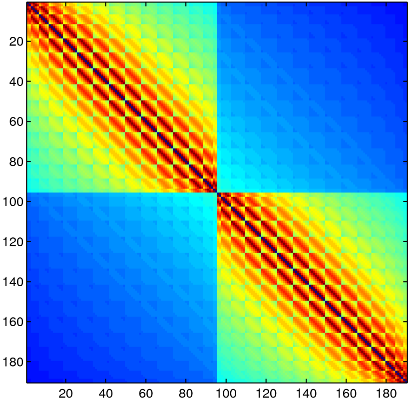


(b) Laplacian Eigenfunctions 10:18

Comparison with PCA: Kernel Matrix

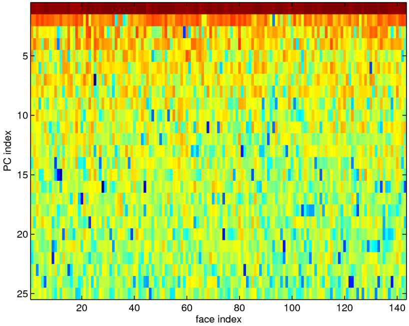


(a) Covariance

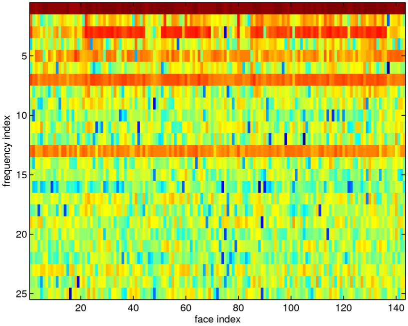


(b) Harmonic kernel

Comparison with PCA: Energy Distribution over Coordinates

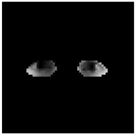


(a) KLB/PCA

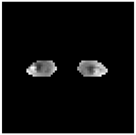


(b) Laplacian Eigenfunctions

Comparison with PCA: Basis Vector #7 . . .

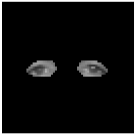


c_7:large

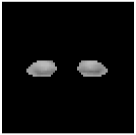


c_7:large

φ_7

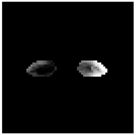


c_7:small

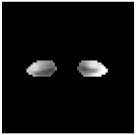


c_7:small

Comparison with PCA: Basis Vector #13 . . .



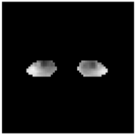
c_{13}:large



c_{13}:large

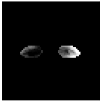
φ_{13}

c_{13}:small

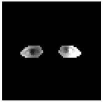


c_{13}:small

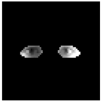
Asymmetry Detector



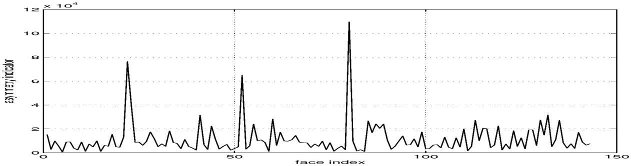
Eyes #80



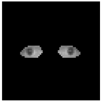
Eyes #22



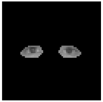
Eyes #52



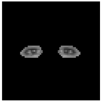
Asymmetry detector



Eyes #5

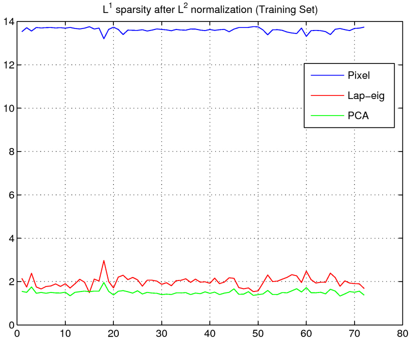


Eyes #84



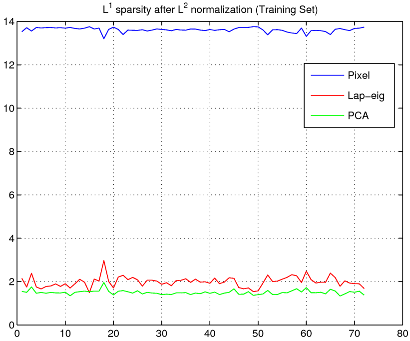
Eyes #59

Comparison with PCA: Sparsity



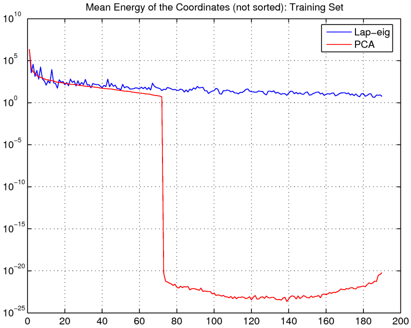
(a) Training set

Comparison with PCA: Sparsity



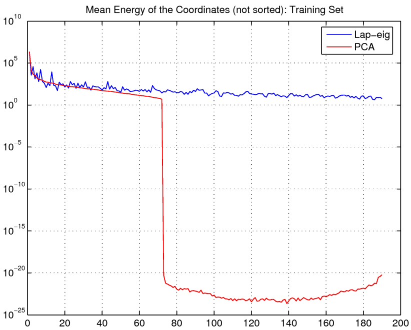
(a) Training set (b) Test set

Comparison with PCA: Coefficient Decay

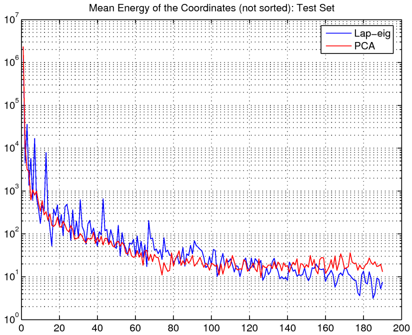


(a) Training set

Comparison with PCA: Coefficient Decay



(a) Training set



(b) Test set

Outline

Solving the Heat Equation on a Complicated Domain

- It is well known that the *semigroup* $\mathrm{e}^{t\Delta}$ can be diagonalized using the Laplacian eigenbasis, i.e., for any initial heat distribution $u_0(\boldsymbol{x}) \in L^2(\overline{\Omega})$, we have the heat distribution at time t formally as

$$u(\boldsymbol{x}, t) = \mathrm{e}^{t\Delta} u_0 = \sum_{j=1}^{\infty} \mathrm{e}^{-t\lambda_j} \langle u_0, \varphi_j \rangle \varphi_j(\boldsymbol{x}),$$

which is based on the expansion of the heat kernel (*Green's function for the heat equation*) $p_t(\boldsymbol{x}, \boldsymbol{y})$ via the Laplacian eigenfunctions as follows:

$$p_t(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{\infty} \mathrm{e}^{-\lambda_j t} \varphi_j(\boldsymbol{x}) \overline{\varphi_j(\boldsymbol{y})} \quad (t, \boldsymbol{x}, \boldsymbol{y}) \in (0, \infty) \times \overline{\Omega} \times \overline{\Omega}.$$

Discretization of the Problem

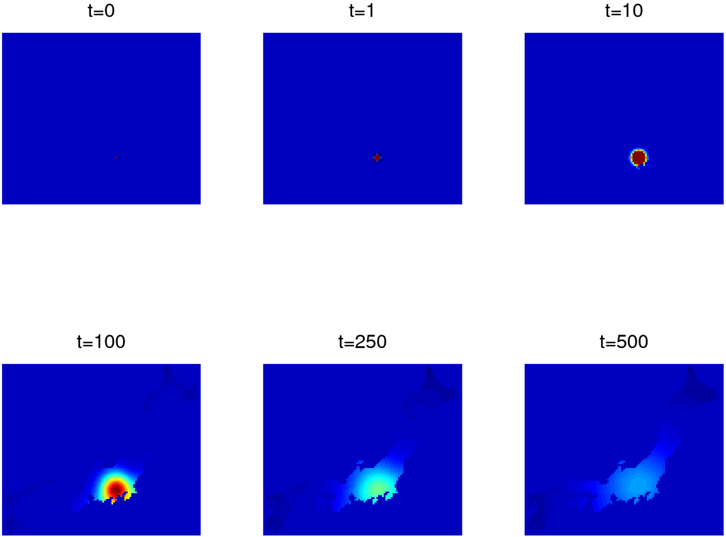
- Due to the discretization of the problem, we can write $e^{t\Delta}$ in the matrix-vector notation as

$$\Phi\, e^{-t\Lambda}\, \Phi^{\mathsf{T}} = \Phi\, \mathrm{diag}\left(e^{-t\lambda_1}, \ldots, e^{-t\lambda_N}\right) \Phi^{\mathsf{T}} = \sum_{j=1}^{N} e^{-\lambda_j t} \boldsymbol{\varphi}_j \boldsymbol{\varphi}_j^{\mathsf{T}},$$

  where $\Phi = \left(\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_N\right)$ is the Laplacian eigenbasis matrix of size $N \times N$, and Λ is the diagonal matrix consisting of the eigenvalues of the Laplacian, which are the inverse of the eigenvalues of the kernel matrix, i.e., $\Lambda_{k,k} = \lambda_k = 1/\mu_k$.
- Given an initial heat distribution over the domain, $\boldsymbol{u}_0 \in \mathbb{R}^N$, we can compute the heat distribution at time t as

$$\boldsymbol{u}(t) = \Phi\, e^{-t\Lambda}\, \Phi^{\mathsf{T}} \boldsymbol{u}_0.$$

Simulation Experiments



Remarks on the Boundary Condition

- It is well known that the eigenvalues of the Laplacian with the Dirichlet (or Neumann) BC are positive (or non-negative, respectively) while the Robin BC could have a *negative* eigenvalue.
- Using our commuting integral operator approach, it is difficult to precisely specify the BC because our formulation satisfies neither the Dirichlet nor the Neumann nor the Robin conditions.
- Our empirical observation so far has led to the following conjecture:

Conjecture (NS 2007)

The eigenvalues of the Laplacian satisfying our BC and defined over a bounded domain $\Omega \subset \mathbb{R}^d$ are all positive possibly with a finite number of negative ones.

Remarks on the Boundary Condition

- It is well known that the eigenvalues of the Laplacian with the Dirichlet (or Neumann) BC are positive (or non-negative, respectively) while the Robin BC could have a *negative* eigenvalue.
- Using our commuting integral operator approach, it is difficult to precisely specify the BC because our formulation satisfies neither the Dirichlet nor the Neumann nor the Robin conditions.
- Our empirical observation so far has led to the following conjecture:

Conjecture (NS 2007)

The eigenvalues of the Laplacian satisfying our BC and defined over a bounded domain $\Omega \in \mathbb{R}^d$ are all positive possibly with a finite number of negative ones.

Remarks on the Boundary Condition

- It is well known that the eigenvalues of the Laplacian with the Dirichlet (or Neumann) BC are positive (or non-negative, respectively) while the Robin BC could have a *negative* eigenvalue.
- Using our commuting integral operator approach, it is difficult to precisely specify the BC because our formulation satisfies neither the Dirichlet nor the Neumann nor the Robin conditions.
- Our empirical observation so far has led to the following conjecture:

Conjecture (NS 2007)

The eigenvalues of the Laplacian satisfying our BC and defined over a bounded domain $\Omega \in \mathbb{R}^d$ are all positive possibly with a finite number of negative ones.

Remarks on the Boundary Condition

- It is well known that the eigenvalues of the Laplacian with the Dirichlet (or Neumann) BC are positive (or non-negative, respectively) while the Robin BC could have a *negative* eigenvalue.
- Using our commuting integral operator approach, it is difficult to precisely specify the BC because our formulation satisfies neither the Dirichlet nor the Neumann nor the Robin conditions.
- Our empirical observation so far has led to the following conjecture:

Conjecture (NS 2007)

The eigenvalues of the Laplacian satisfying our BC and defined over a bounded domain $\Omega \in \mathbb{R}^d$ are all positive possibly with a finite number of negative ones.

Outline

What Are Patient-Specific Basis Functions?

- Proposed first by D. W. Winters et al.: "Three-dimensional microwave breast imaging: Dispersive dielectric properties estimation using patient-specific basis functions," *IEEE Trans. Medical Imaging*, vol. 28, no. 7, pp. 969–981, 2009.
- Objective: Speed up the imaging process of a Region Of Interest (ROI) in microwave breast imaging.
- Idea: Represent an ROI by a linear combination of a small number of the flexible basis functions adapted to individual patients $\Longrightarrow$ more computationally efficient than voxel-based representations.
- First I will explain their method using a 1D model for simplicity (their actual 3D model is simply a tensor product of the 1D model), and give my own interpretation: their method is essentially equivalent to *computing the Karhunen-Loève Transform assuming the autocorrelation function over I is Gaussian*.
- Then, I will discuss the potential problems of this approach.

What Are Patient-Specific Basis Functions?

- Proposed first by D. W. Winters et al.: "Three-dimensional microwave breast imaging: Dispersive dielectric properties estimation using patient-specific basis functions," *IEEE Trans. Medical Imaging*, vol. 28, no. 7, pp. 969–981, 2009.
- Objective: Speed up the imaging process of a Region Of Interest (ROI) in microwave breast imaging.
- Idea: Represent an ROI by a linear combination of a small number of the flexible basis functions adapted to individual patients $\Longrightarrow$ more computationally efficient than voxel-based representations.
- First I will explain their method using a 1D model for simplicity (their actual 3D model is simply a tensor product of the 1D model), and give my own interpretation: their method is essentially equivalent to *computing the Karhunen-Loève Transform assuming the autocorrelation function over I is Gaussian*.
- Then, I will discuss the potential problems of this approach.

What Are Patient-Specific Basis Functions?

- Proposed first by D. W. Winters et al.: "Three-dimensional microwave breast imaging: Dispersive dielectric properties estimation using patient-specific basis functions," *IEEE Trans. Medical Imaging*, vol. 28, no. 7, pp. 969–981, 2009.
- Objective: Speed up the imaging process of a Region Of Interest (ROI) in microwave breast imaging.
- Idea: Represent an ROI by a linear combination of a small number of the flexible basis functions adapted to individual patients $\Longrightarrow$ more computationally efficient than voxel-based representations.
- First I will explain their method using a 1D model for simplicity (their actual 3D model is simply a tensor product of the 1D model), and give my own interpretation: their method is essentially equivalent to *computing the Karhunen-Loève Transform assuming the autocorrelation function over I is Gaussian.*
- Then, I will discuss the potential problems of this approach.

What Are Patient-Specific Basis Functions?

- Proposed first by D. W. Winters et al.: "Three-dimensional microwave breast imaging: Dispersive dielectric properties estimation using patient-specific basis functions," *IEEE Trans. Medical Imaging*, vol. 28, no. 7, pp. 969–981, 2009.
- Objective: Speed up the imaging process of a Region Of Interest (ROI) in microwave breast imaging.
- Idea: Represent an ROI by a linear combination of a small number of the flexible basis functions adapted to individual patients $\Longrightarrow$ more computationally efficient than voxel-based representations.
- First I will explain their method using a 1D model for simplicity (their actual 3D model is simply a tensor product of the 1D model), and give my own interpretation: their method is essentially equivalent to *computing the Karhunen-Loève Transform assuming the autocorrelation function over I is Gaussian*.
- Then, I will discuss the potential problems of this approach.

What Are Patient-Specific Basis Functions?

- Proposed first by D. W. Winters et al.: "Three-dimensional microwave breast imaging: Dispersive dielectric properties estimation using patient-specific basis functions," *IEEE Trans. Medical Imaging*, vol. 28, no. 7, pp. 969–981, 2009.
- Objective: Speed up the imaging process of a Region Of Interest (ROI) in microwave breast imaging.
- Idea: Represent an ROI by a linear combination of a small number of the flexible basis functions adapted to individual patients $\Longrightarrow$ more computationally efficient than voxel-based representations.
- First I will explain their method using a 1D model for simplicity (their actual 3D model is simply a tensor product of the 1D model), and give my own interpretation: their method is essentially equivalent to *computing the Karhunen-Loève Transform assuming the autocorrelation function over I is Gaussian*.
- Then, I will discuss the potential problems of this approach.

Patient-Specific Basis Functions . . .

- Let Ω be an ROI, which is a subset of $I := [0,1]$.
- Suppose we discretize I into N cells (or bins) whose centers are $x_k = (k - 1/2)/N$, $k = 1, \dots, N$.
- Let $\sigma = 0.75 * |I|/N$, and consider a set of *shifted Gaussian functions*,

$$g_k(x \mid \sigma) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - x_k)^2}{2\sigma^2}\right) \quad x \in I.$$

- Construct a matrix $G \in \mathbb{R}^{N \times N}$ where kth column vector is $\boldsymbol{g}_k = \left(g_k(x_1 \mid \sigma), g_k(x_2 \mid \sigma), \cdots, g_k(x_N \mid \sigma)\right)^{\mathsf{T}}$.
- Suppose $\Omega = \{x_{k_0}, x_{k_0+1}, \dots, x_{k_1}\} \subset I$, $|\Omega| = k_1 - k_0 + 1$, and let us define the normalized discrete characteristic function $\chi_\Omega \in \mathbb{R}^N$:

$$\chi_\Omega(k) := \begin{cases} \frac{1}{\sqrt{|\Omega|}} & \text{if } k_0 \le k \le k_1; \\ 0 & \text{otherwise.} \end{cases}$$

- Keep χ_Ω as the basis vector for the *DC component*, and consider the truncated matrix $G_\Omega := \left[\chi_\Omega .* \boldsymbol{g}_1 \mid \chi_\Omega .* \boldsymbol{g}_2 \mid \cdots \mid \chi_\Omega .* \boldsymbol{g}_N\right] \in \mathbb{R}^{N \times N}$.

Patient-Specific Basis Functions ...

- Let Ω be an ROI, which is a subset of $I := [0, 1]$.
- Suppose we discretize I into N cells (or bins) whose centers are $x_k = (k - 1/2)/N$, $k = 1, \ldots, N$.
- Let $\sigma = 0.75 * |I|/N$, and consider a set of *shifted Gaussian functions*,

$$g_k(x \mid \sigma) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - x_k)^2}{2\sigma^2}\right) \quad x \in I.$$

- Construct a matrix $G \in \mathbb{R}^{N \times N}$ where kth column vector is $\boldsymbol{g}_k = \left(g_k(x_1 \mid \sigma), g_k(x_2 \mid \sigma), \cdots, g_k(x_N \mid \sigma)\right)^{\mathsf{T}}$.
- Suppose $\Omega = \{x_{k_0}, x_{k_0+1}, \ldots, x_{k_1}\} \subset I$, $|\Omega| = k_1 - k_0 + 1$, and let us define the normalized discrete characteristic function $\chi_\Omega \in \mathbb{R}^N$:

$$\chi_\Omega(k) := \begin{cases} \frac{1}{\sqrt{|\Omega|}} & \text{if } k_0 \le k \le k_1; \\ 0 & \text{otherwise.} \end{cases}$$

- Keep χ_Ω as the basis vector for the *DC component*, and consider the truncated matrix $G_\Omega := \left[\chi_\Omega .* \boldsymbol{g}_1 \mid \chi_\Omega .* \boldsymbol{g}_2 \mid \cdots \mid \chi_\Omega .* \boldsymbol{g}_N\right] \in \mathbb{R}^{N \times N}$.

Patient-Specific Basis Functions ...

- Let Ω be an ROI, which is a subset of $I := [0,1]$.
- Suppose we discretize I into N cells (or bins) whose centers are $x_k = (k - 1/2)/N, \ k = 1, \ldots, N$.
- Let $\sigma = 0.75 * |I|/N$, and consider a set of *shifted Gaussian functions*,

$$g_k(x \mid \sigma) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - x_k)^2}{2\sigma^2}\right) \quad x \in I.$$

- Construct a matrix $G \in \mathbb{R}^{N \times N}$ where kth column vector is $\boldsymbol{g}_k = \left(g_k(x_1 \mid \sigma), g_k(x_2 \mid \sigma), \cdots, g_k(x_N \mid \sigma)\right)^{\mathsf{T}}$.
- Suppose $\Omega = \{x_{k_0}, x_{k_0+1}, \ldots, x_{k_1}\} \subset I$, $|\Omega| = k_1 - k_0 + 1$, and let us define the normalized discrete characteristic function $\chi_\Omega \in \mathbb{R}^N$:

$$\chi_\Omega(k) := \begin{cases} \frac{1}{\sqrt{|\Omega|}} & \text{if } k_0 \le k \le k_1; \\ 0 & \text{otherwise.} \end{cases}$$

- Keep χ_Ω as the basis vector for the *DC component*, and consider the truncated matrix $G_\Omega := \left[\chi_\Omega .* \boldsymbol{g}_1 \mid \chi_\Omega .* \boldsymbol{g}_2 \mid \cdots \mid \chi_\Omega .* \boldsymbol{g}_N\right] \in \mathbb{R}^{N \times N}$.

Patient-Specific Basis Functions . . .

- Let Ω be an ROI, which is a subset of $I := [0,1]$.
- Suppose we discretize I into N cells (or bins) whose centers are $x_k = (k - 1/2)/N,\ k = 1, \ldots, N$.
- Let $\sigma = 0.75 * |I|/N$, and consider a set of *shifted Gaussian functions*,

$$g_k(x \mid \sigma) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - x_k)^2}{2\sigma^2}\right) \quad x \in I.$$

- Construct a matrix $G \in \mathbb{R}^{N \times N}$ where kth column vector is $\boldsymbol{g}_k = \left(g_k(x_1 \mid \sigma), g_k(x_2 \mid \sigma), \cdots, g_k(x_N \mid \sigma)\right)^{\mathsf{T}}$.
- Suppose $\Omega = \{x_{k_0}, x_{k_0+1}, \ldots, x_{k_1}\} \subset I$, $|\Omega| = k_1 - k_0 + 1$, and let us define the normalized discrete characteristic function $\chi_\Omega \in \mathbb{R}^N$:

$$\chi_\Omega(k) := \begin{cases} \frac{1}{\sqrt{|\Omega|}} & \text{if } k_0 \le k \le k_1; \\ 0 & \text{otherwise.} \end{cases}$$

- Keep χ_Ω as the basis vector for the *DC component*, and consider the truncated matrix $G_\Omega := \left[\chi_\Omega .* \boldsymbol{g}_1 \mid \chi_\Omega .* \boldsymbol{g}_2 \mid \cdots \mid \chi_\Omega .* \boldsymbol{g}_N\right] \in \mathbb{R}^{N \times N}$.

Patient-Specific Basis Functions . . .

- Let Ω be an ROI, which is a subset of $I := [0,1]$.
- Suppose we discretize I into N cells (or bins) whose centers are $x_k = (k - 1/2)/N$, $k = 1, \ldots, N$.
- Let $\sigma = 0.75 * |I|/N$, and consider a set of *shifted Gaussian functions*,

$$g_k(x \mid \sigma) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - x_k)^2}{2\sigma^2}\right) \quad x \in I.$$

- Construct a matrix $G \in \mathbb{R}^{N \times N}$ where kth column vector is $\boldsymbol{g}_k = \left(g_k(x_1 \mid \sigma), g_k(x_2 \mid \sigma), \cdots, g_k(x_N \mid \sigma)\right)^{\mathsf{T}}$.
- Suppose $\Omega = \{x_{k_0}, x_{k_0+1}, \ldots, x_{k_1}\} \subset I$, $|\Omega| = k_1 - k_0 + 1$, and let us define the normalized discrete characteristic function $\boldsymbol{\chi}_\Omega \in \mathbb{R}^N$:

$$\chi_\Omega(k) := \begin{cases} \frac{1}{\sqrt{|\Omega|}} & \text{if } k_0 \le k \le k_1; \\ 0 & \text{otherwise.} \end{cases}$$

- Keep χ_Ω as the basis vector for the *DC component*, and consider the truncated matrix $G_\Omega := [\chi_\Omega .* \boldsymbol{g}_1 \mid \chi_\Omega .* \boldsymbol{g}_2 \mid \cdots \mid \chi_\Omega .* \boldsymbol{g}_N] \in \mathbb{R}^{N \times N}$.

Patient-Specific Basis Functions . . .

- Let Ω be an ROI, which is a subset of $I := [0, 1]$.
- Suppose we discretize I into N cells (or bins) whose centers are $x_k = (k - 1/2)/N$, $k = 1, \ldots, N$.
- Let $\sigma = 0.75 * |I|/N$, and consider a set of *shifted Gaussian functions*,

$$g_k(x \mid \sigma) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - x_k)^2}{2\sigma^2}\right) \quad x \in I.$$

- Construct a matrix $G \in \mathbb{R}^{N \times N}$ where kth column vector is $\boldsymbol{g}_k = \left(g_k(x_1 \mid \sigma), g_k(x_2 \mid \sigma), \cdots, g_k(x_N \mid \sigma)\right)^{\mathsf{T}}$.
- Suppose $\Omega = \{x_{k_0}, x_{k_0+1}, \ldots, x_{k_1}\} \subset I$, $|\Omega| = k_1 - k_0 + 1$, and let us define the normalized discrete characteristic function $\boldsymbol{\chi}_\Omega \in \mathbb{R}^N$:

$$\chi_\Omega(k) := \begin{cases} \frac{1}{\sqrt{|\Omega|}} & \text{if } k_0 \le k \le k_1; \\ 0 & \text{otherwise.} \end{cases}$$

- Keep $\boldsymbol{\chi}_\Omega$ as the basis vector for the *DC component*, and consider the truncated matrix $G_\Omega := \left[\boldsymbol{\chi}_\Omega .* \boldsymbol{g}_1 \mid \boldsymbol{\chi}_\Omega .* \boldsymbol{g}_2 \mid \cdots \mid \boldsymbol{\chi}_\Omega .* \boldsymbol{g}_N\right] \in \mathbb{R}^{N \times N}$.

Patient-Specific Basis Functions . . .

- Then, consider the *orthogonal complement* to the 1D subspace $\mathrm{span}\{\boldsymbol{\chi}_\Omega\}$ in $\mathbb{R}^N$:
$$\widetilde{G}_\Omega = \left(I - \boldsymbol{\chi}_\Omega \boldsymbol{\chi}_\Omega^{\mathsf{T}}\right) G_\Omega.$$
- The *Singular Value Decomposition* (SVD) of $\widetilde{G}_\Omega$ is computed, i.e., $\widetilde{G}_\Omega = U\Sigma V^{\mathsf{T}}$.
- Finally, Winters et al. suggest that a small number, say $\ell(\ll N)$, of column vectors of U to represent an object on Ω approximately.
- Suppose the original imaging system equation be written as $A\boldsymbol{x} = \boldsymbol{b}$ where $A \in \mathbb{R}^{m\times N}$ is a imaging system matrix, $\boldsymbol{x} \in \mathbb{R}^N$ is the object values over I, and $\boldsymbol{b} \in \mathbb{R}^m$ is the measured data.
- Let $U_\ell \in \mathbb{R}^{N\times\ell} := [\boldsymbol{\chi}_\Omega, U(:, 1 : \ell - 1)]$. (Note $U_\ell^{\mathsf{T}} U_\ell = I_\ell$.) Then, Winters et al. suggest approximating $\boldsymbol{x}$ using the ℓ basis vectors (i.e., column vectors) of U_ℓ, i.e., $\boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell$ and solving for $\widetilde{\boldsymbol{x}}_\ell \in \mathbb{R}^\ell$:
$$AU_\ell \widetilde{\boldsymbol{x}}_\ell = \boldsymbol{b}, \quad \boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell.$$

Patient-Specific Basis Functions . . .

- Then, consider the *orthogonal complement* to the 1D subspace $\operatorname{span}\{\boldsymbol{\chi}_\Omega\}$ in $\mathbb{R}^N$:
$$\widetilde{G}_\Omega = \left(I - \boldsymbol{\chi}_\Omega \boldsymbol{\chi}_\Omega^{\mathsf{T}}\right) G_\Omega.$$
- The *Singular Value Decomposition* (SVD) of $\widetilde{G}_\Omega$ is computed, i.e., $\widetilde{G}_\Omega = U \Sigma V^{\mathsf{T}}$.
- Finally, Winters et al. suggest that a small number, say $\ell (\ll N)$, of column vectors of U to represent an object on Ω approximately.
- Suppose the original imaging system equation be written as $A\boldsymbol{x} = \boldsymbol{b}$ where $A \in \mathbb{R}^{m \times N}$ is a imaging system matrix, $\boldsymbol{x} \in \mathbb{R}^N$ is the object values over I, and $\boldsymbol{b} \in \mathbb{R}^m$ is the measured data.
- Let $U_\ell \in \mathbb{R}^{N \times \ell} := [\boldsymbol{\chi}_\Omega, U(:, 1 : \ell - 1)]$. (Note $U_\ell^{\mathsf{T}} U_\ell = I_\ell$.) Then, Winters et al. suggest approximating $\boldsymbol{x}$ using the ℓ basis vectors (i.e., column vectors) of U_ℓ, i.e., $\boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell$ and solving for $\widetilde{\boldsymbol{x}}_\ell \in \mathbb{R}^\ell$:
$$A U_\ell \widetilde{\boldsymbol{x}}_\ell = \boldsymbol{b}, \quad \boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell.$$

Patient-Specific Basis Functions . . .

- Then, consider the *orthogonal complement* to the 1D subspace $\mathrm{span}\{\boldsymbol{\chi}_\Omega\}$ in $\mathbb{R}^N$:
$$\widetilde{G}_\Omega = \left(I - \boldsymbol{\chi}_\Omega \boldsymbol{\chi}_\Omega^\mathsf{T}\right) G_\Omega.$$
- The *Singular Value Decomposition* (SVD) of $\widetilde{G}_\Omega$ is computed, i.e., $\widetilde{G}_\Omega = U\Sigma V^\mathsf{T}$.
- Finally, Winters et al. suggest that a small number, say $\ell (\ll N)$, of column vectors of U to represent an object on Ω approximately.
- Suppose the original imaging system equation be written as $A\boldsymbol{x} = \boldsymbol{b}$ where $A \in \mathbb{R}^{m\times N}$ is a imaging system matrix, $\boldsymbol{x} \in \mathbb{R}^N$ is the object values over I, and $\boldsymbol{b} \in \mathbb{R}^m$ is the measured data.
- Let $U_\ell \in \mathbb{R}^{N\times \ell} := \left[\boldsymbol{\chi}_\Omega,\ U(:,1:\ell-1)\right]$. (Note $U_\ell^\mathsf{T} U_\ell = I_\ell$.) Then, Winters et al. suggest approximating $\boldsymbol{x}$ using the ℓ basis vectors (i.e., column vectors) of U_ℓ, i.e., $\boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell$ and solving for $\widetilde{\boldsymbol{x}}_\ell \in \mathbb{R}^\ell$:
$$AU_\ell \widetilde{\boldsymbol{x}}_\ell = \boldsymbol{b}, \quad \boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell.$$

Patient-Specific Basis Functions . . .

- Then, consider the *orthogonal complement* to the 1D subspace $\mathrm{span}\{\boldsymbol{\chi}_\Omega\}$ in $\mathbb{R}^N$:
$$\widetilde{G}_\Omega = \left(I - \boldsymbol{\chi}_\Omega \boldsymbol{\chi}_\Omega^\mathsf{T}\right) G_\Omega.$$
- The *Singular Value Decomposition* (SVD) of $\widetilde{G}_\Omega$ is computed, i.e., $\widetilde{G}_\Omega = U\Sigma V^\mathsf{T}$.
- Finally, Winters et al. suggest that a small number, say $\ell(\ll N)$, of column vectors of U to represent an object on Ω approximately.
- Suppose the original imaging system equation be written as $A\boldsymbol{x} = \boldsymbol{b}$ where $A \in \mathbb{R}^{m\times N}$ is a imaging system matrix, $\boldsymbol{x} \in \mathbb{R}^N$ is the object values over I, and $\boldsymbol{b} \in \mathbb{R}^m$ is the measured data.
- Let $U_\ell \in \mathbb{R}^{N\times\ell} := [\chi_\Omega, U(:,1:\ell-1)]$. (Note $U_\ell^\mathsf{T} U_\ell = I_\ell$.) Then, Winters et al. suggest approximating $\boldsymbol{x}$ using the ℓ basis vectors (i.e., column vectors) of U_ℓ, i.e., $\boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell$ and solving for $\widetilde{\boldsymbol{x}}_\ell \in \mathbb{R}^\ell$:
$$AU_\ell\widetilde{\boldsymbol{x}}_\ell = \boldsymbol{b}, \quad \boldsymbol{x} \approx U_\ell\widetilde{\boldsymbol{x}}_\ell.$$

Patient-Specific Basis Functions . . .

- Then, consider the *orthogonal complement* to the 1D subspace $\mathrm{span}\{\boldsymbol{\chi}_\Omega\}$ in $\mathbb{R}^N$:
$$\widetilde{G}_\Omega = \left(I - \boldsymbol{\chi}_\Omega \boldsymbol{\chi}_\Omega^\mathsf{T}\right) G_\Omega.$$
- The *Singular Value Decomposition* (SVD) of $\widetilde{G}_\Omega$ is computed, i.e., $\widetilde{G}_\Omega = U\Sigma V^\mathsf{T}$.
- Finally, Winters et al. suggest that a small number, say $\ell(\ll N)$, of column vectors of U to represent an object on Ω approximately.
- Suppose the original imaging system equation be written as $A\boldsymbol{x} = \boldsymbol{b}$ where $A \in \mathbb{R}^{m\times N}$ is a imaging system matrix, $\boldsymbol{x} \in \mathbb{R}^N$ is the object values over I, and $\boldsymbol{b} \in \mathbb{R}^m$ is the measured data.
- Let $U_\ell \in \mathbb{R}^{N\times\ell} := \left[\boldsymbol{\chi}_\Omega,\ U(:,1:\ell-1)\right]$. (Note $U_\ell^\mathsf{T} U_\ell = I_\ell$.) Then, Winters et al. suggest approximating $\boldsymbol{x}$ using the ℓ basis vectors (i.e., column vectors) of U_ℓ, i.e., $\boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell$ and solving for $\widetilde{\boldsymbol{x}}_\ell \in \mathbb{R}^\ell$:
$$AU_\ell \widetilde{\boldsymbol{x}}_\ell = \boldsymbol{b}, \quad \boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell.$$

Patient-Specific Basis Functions ... My Interpretation

- The accuracy and efficiency of the above procedure strongly depends on the quality of the ℓ-term approximation $\boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell$, i.e., ℓ is a tradeoff parameter.
- Going back to the SVD of $\widetilde{G}_\Omega$, U is the solution to the eigenvalue problem of $\widetilde{G}_\Omega \widetilde{G}_\Omega^{\mathsf{T}} U = U \Sigma^2$.
- This means that the columns of U form the basis of the KLT assuming that the underlying autocovariance matrix is $\widetilde{G}_\Omega \widetilde{G}_\Omega^{\mathsf{T}}$.
- The corresponding autocorrelation matrix is $G_\Omega G_\Omega^{\mathsf{T}}$, and this implies that we can view the whole process as an analysis of the following stochastic process in $\mathbb{R}^N$: *Pick uniformly randomly* $x_k \in \Omega$ *and generate a shifted and truncated Gaussian vector* $\chi_\Omega .* \boldsymbol{g}_k$.
- Since each realization is a shifted version of a single vector followed by truncation, we can show that the corresponding KLT/PCA basis are essentially the Discrete Fourier Sine basis supported on Ω. More precisely, they are adjusted versions of DST basis orthogonal to the constant DC component χ_Ω.

Patient-Specific Basis Functions . . . My Interpretation

- The accuracy and efficiency of the above procedure strongly depends on the quality of the ℓ-term approximation $\boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell$, i.e., ℓ is a tradeoff parameter.
- Going back to the SVD of $\widetilde{G}_\Omega$, U is the solution to the eigenvalue problem of $\widetilde{G}_\Omega \widetilde{G}_\Omega^{\mathsf{T}} U = U\Sigma^2$.
- This means that the columns of U form the basis of the KLT assuming that the underlying autocovariance matrix is $\widetilde{G}_\Omega \widetilde{G}_\Omega^{\mathsf{T}}$.
- The corresponding autocorrelation matrix is $G_\Omega G_\Omega^{\mathsf{T}}$, and this implies that we can view the whole process as an analysis of the following stochastic process in $\mathbb{R}^N$: *Pick uniformly randomly $x_k \in \Omega$ and generate a shifted and truncated Gaussian vector $\chi_\Omega .* \boldsymbol{g}_k$.*
- Since each realization is a shifted version of a single vector followed by truncation, we can show that the corresponding KLT/PCA basis are essentially the Discrete Fourier Sine basis supported on Ω. More precisely, they are adjusted versions of DST basis orthogonal to the constant DC component χ_Ω.

Patient-Specific Basis Functions . . . My Interpretation

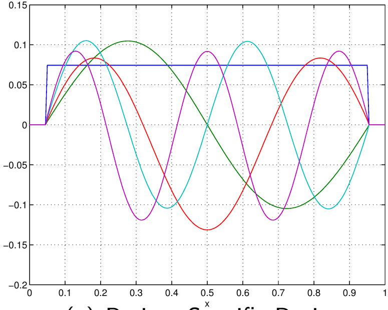
- The accuracy and efficiency of the above procedure strongly depends on the quality of the ℓ-term approximation $\boldsymbol{x} \approx U_\ell \tilde{\boldsymbol{x}}_\ell$, i.e., ℓ is a tradeoff parameter.
- Going back to the SVD of $\widetilde{G}_\Omega$, U is the solution to the eigenvalue problem of $\widetilde{G}_\Omega \widetilde{G}_\Omega^{\mathsf{T}} U = U\Sigma^2$.
- This means that the columns of U form the basis of the KLT assuming that the underlying autocovariance matrix is $\widetilde{G}_\Omega \widetilde{G}_\Omega^{\mathsf{T}}$.
- The corresponding autocorrelation matrix is $G_\Omega G_\Omega^{\mathsf{T}}$, and this implies that we can view the whole process as an analysis of the following stochastic process in $\mathbb{R}^N$: *Pick uniformly randomly $x_k \in \Omega$ and generate a shifted and truncated Gaussian vector $\chi_\Omega .* \boldsymbol{g}_k$.*
- Since each realization is a shifted version of a single vector followed by truncation, we can show that the corresponding KLT/PCA basis are essentially the Discrete Fourier Sine basis supported on Ω. More precisely, they are adjusted versions of DST basis orthogonal to the constant DC component χ_Ω.

Patient-Specific Basis Functions . . . My Interpretation

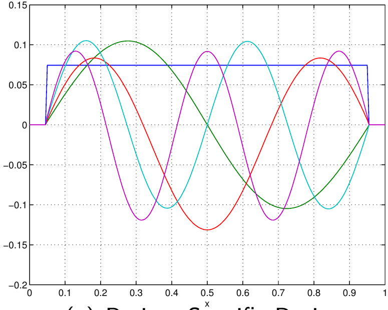
- The accuracy and efficiency of the above procedure strongly depends on the quality of the ℓ-term approximation $\boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell$, i.e., ℓ is a tradeoff parameter.
- Going back to the SVD of $\widetilde{G}_\Omega$, U is the solution to the eigenvalue problem of $\widetilde{G}_\Omega \widetilde{G}_\Omega^{\mathsf{T}} U = U \Sigma^2$.
- This means that the columns of U form the basis of the KLT assuming that the underlying autocovariance matrix is $\widetilde{G}_\Omega \widetilde{G}_\Omega^{\mathsf{T}}$.
- The corresponding autocorrelation matrix is $G_\Omega G_\Omega^{\mathsf{T}}$, and this implies that we can view the whole process as an analysis of the following stochastic process in $\mathbb{R}^N$: *Pick uniformly randomly $x_k \in \Omega$ and generate a shifted and truncated Gaussian vector $\boldsymbol{\chi}_\Omega .* \boldsymbol{g}_k$.*
- Since each realization is a shifted version of a single vector followed by truncation, we can show that the corresponding KLT/PCA basis are essentially the Discrete Fourier Sine basis supported on Ω. More precisely, they are adjusted versions of DST basis orthogonal to the constant DC component χ_Ω.

Patient-Specific Basis Functions . . . My Interpretation

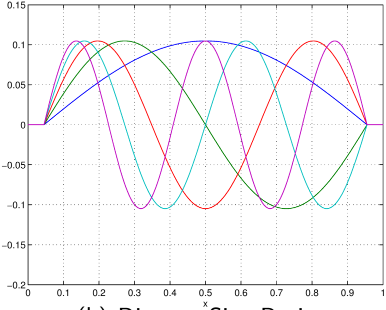
- The accuracy and efficiency of the above procedure strongly depends on the quality of the ℓ-term approximation $\boldsymbol{x} \approx U_\ell \widetilde{\boldsymbol{x}}_\ell$, i.e., ℓ is a tradeoff parameter.
- Going back to the SVD of $\widetilde{G}_\Omega$, U is the solution to the eigenvalue problem of $\widetilde{G}_\Omega \widetilde{G}_\Omega^\mathsf{T} U = U\Sigma^2$.
- This means that the columns of U form the basis of the KLT assuming that the underlying autocovariance matrix is $\widetilde{G}_\Omega \widetilde{G}_\Omega^\mathsf{T}$.
- The corresponding autocorrelation matrix is $G_\Omega G_\Omega^\mathsf{T}$, and this implies that we can view the whole process as an analysis of the following stochastic process in $\mathbb{R}^N$: *Pick uniformly randomly $x_k \in \Omega$ and generate a shifted and truncated Gaussian vector $\boldsymbol{\chi}_\Omega .* \boldsymbol{g}_k$.*
- Since each realization is a shifted version of a single vector followed by truncation, we can show that the corresponding KLT/PCA basis are essentially the Discrete Fourier Sine basis supported on Ω. More precisely, they are adjusted versions of DST basis orthogonal to the constant DC component $\boldsymbol{\chi}_\Omega$.



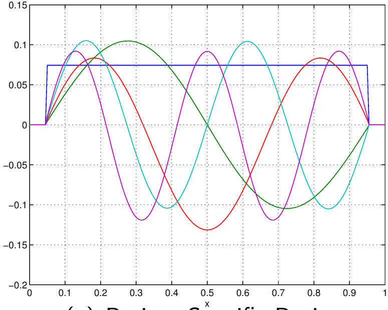
(a) Patient-Specific Basis



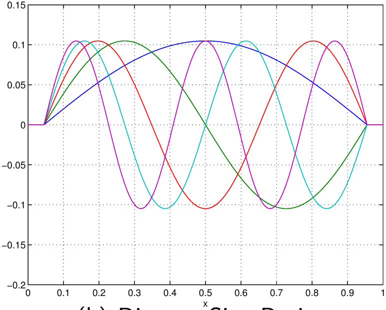
(a) Patient-Specific Basis



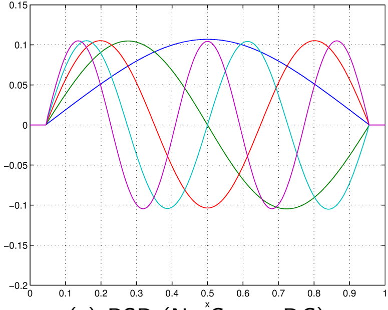
(b) Discrete Sine Basis



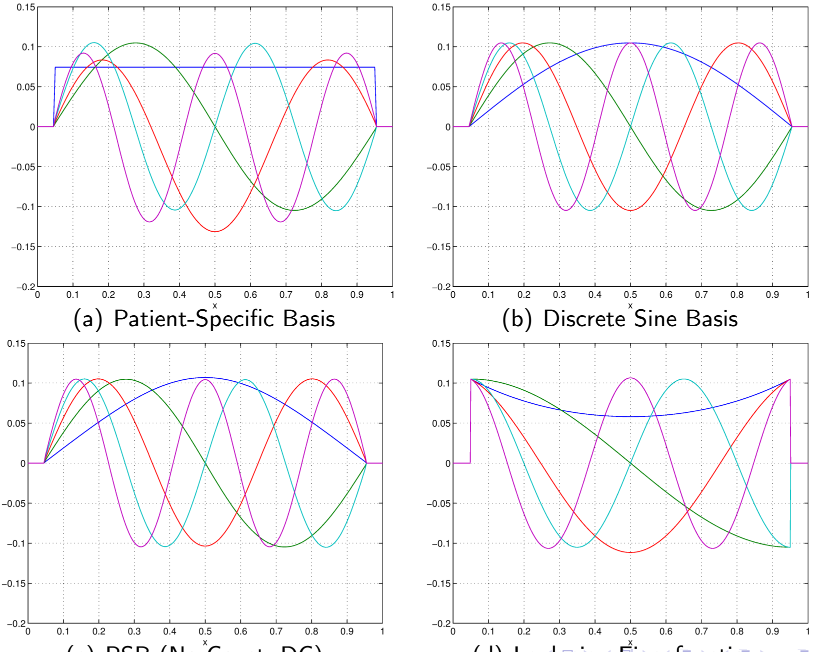
(a) Patient-Specific Basis



(b) Discrete Sine Basis



(c) PSB (No Const. DC)



(a) Patient-Specific Basis

(b) Discrete Sine Basis

(c) PSB (No Const. DC)

(d) Laplacian Eigenfunctions

Patient-Specific Basis Functions . . . My Interpretation

- PSB (Patient-Specific Basis)

  Pro: The constant DC component vector χ_Ω is included.

  Con 1: *Features near from the boundary of* Ω may not be represented well with a small number of ℓ due to the Dirichlet BC implicitly imposed by χ_Ω.

  Con 2: In reality, building a basis for a complicated 3D shape based on the *tensor products* may not be easy, and the boundary effects may become more pronounced.

- LE-CI (Laplacian Eigenfunctions via Commuting Integral Operator)

  Pro 1: Features near from the boundary may be more efficiently represented thanks to the more natural BC.

  Pro 2: Building a basis for even a complicated 3D shape is easy; we only need pairwise distances between voxel centers.

  Con: χ_Ω is not included. However, if we wish, we can include χ_Ω by projecting the kernel matrix K onto the orthogonal complement to span χ_Ω before diagonalizing K.

Patient-Specific Basis Functions . . . My Interpretation

- PSB (Patient-Specific Basis)
  - Pro: The constant DC component vector χ_Ω is included.
  - Con 1: *Features near from the boundary of* Ω may not be represented well with a small number of ℓ due to the Dirichlet BC implicitly imposed by χ_Ω.
  - Con 2: In reality, building a basis for a complicated 3D shape based on the *tensor products* may not be easy, and the boundary effects may become more pronounced.
- LE-CI (Laplacian Eigenfunctions via Commuting Integral Operator)
  - Pro 1: Features near from the boundary may be more efficiently represented thanks to the more natural BC.
  - Pro 2: Building a basis for even a complicated 3D shape is easy; we only need pairwise distances between voxel centers.
  - Con: χ_Ω is not included. However, if we wish, we can include χ_Ω by projecting the kernel matrix K onto the orthogonal complement to span χ_Ω before diagonalizing K.

Patient-Specific Basis Functions . . . My Interpretation

- PSB (Patient-Specific Basis)
  - Pro: The constant DC component vector χ_Ω is included.
  - Con 1: *Features near from the boundary of* Ω may not be represented well with a small number of ℓ due to the Dirichlet BC implicitly imposed by χ_Ω.
  - Con 2: In reality, building a basis for a complicated 3D shape based on the *tensor products* may not be easy, and the boundary effects may become more pronounced.
- LE-CI (Laplacian Eigenfunctions via Commuting Integral Operator)
  - Pro 1: Features near from the boundary may be more efficiently represented thanks to the more natural BC.
  - Pro 2: Building a basis for even a complicated 3D shape is easy; we only need pairwise distances between voxel centers.
  - Con: χ_Ω is not included. However, if we wish, we can include χ_Ω by projecting the kernel matrix K onto the orthogonal complement to span χ_Ω before diagonalizing K.

Patient-Specific Basis Functions . . . My Interpretation

- PSB (Patient-Specific Basis)
  - Pro: The constant DC component vector $\boldsymbol{\chi}_\Omega$ is included.
  - Con 1: *Features near from the boundary of* Ω may not be represented well with a small number of ℓ due to the Dirichlet BC implicitly imposed by $\boldsymbol{\chi}_\Omega$.
  - Con 2: In reality, building a basis for a complicated 3D shape based on the *tensor products* may not be easy, and the boundary effects may become more pronounced.
- LE-CI (Laplacian Eigenfunctions via Commuting Integral Operator)
  - Pro 1: Features near from the boundary may be more efficiently represented thanks to the more natural BC.
  - Pro 2: Building a basis for even a complicated 3D shape is easy; we only need pairwise distances between voxel centers.
  - Con: χ_Ω is not included. However, if we wish, we can include χ_Ω by projecting the kernel matrix K onto the orthogonal complement to span χ_Ω before diagonalizing K.

Patient-Specific Basis Functions . . . My Interpretation

- PSB (Patient-Specific Basis)
  - Pro: The constant DC component vector $\boldsymbol{\chi}_\Omega$ is included.
  - Con 1: *Features near from the boundary of* Ω may not be represented well with a small number of ℓ due to the Dirichlet BC implicitly imposed by $\boldsymbol{\chi}_\Omega$.
  - Con 2: In reality, building a basis for a complicated 3D shape based on the *tensor products* may not be easy, and the boundary effects may become more pronounced.
- LE-CI (Laplacian Eigenfunctions via Commuting Integral Operator)
  - Pro 1: Features near from the boundary may be more efficiently represented thanks to the more natural BC.
  - Pro 2: Building a basis for even a complicated 3D shape is easy; we only need pairwise distances between voxel centers.
  - Con: χ_Ω is not included. However, if we wish, we can include χ_Ω by projecting the kernel matrix K onto the orthogonal complement to span χ_Ω before diagonalizing K.

Patient-Specific Basis Functions . . . My Interpretation

- PSB (Patient-Specific Basis)
  - Pro: The constant DC component vector $\boldsymbol{\chi}_\Omega$ is included.
  - Con 1: *Features near from the boundary of* Ω may not be represented well with a small number of ℓ due to the Dirichlet BC implicitly imposed by $\boldsymbol{\chi}_\Omega$.
  - Con 2: In reality, building a basis for a complicated 3D shape based on the *tensor products* may not be easy, and the boundary effects may become more pronounced.
- LE-CI (Laplacian Eigenfunctions via Commuting Integral Operator)
  - Pro 1: Features near from the boundary may be more efficiently represented thanks to the more natural BC.
  - Pro 2: Building a basis for even a complicated 3D shape is easy; we only need pairwise distances between voxel centers.
  - Con: χ_Ω is not included. However, if we wish, we can include χ_Ω by projecting the kernel matrix K onto the orthogonal complement to span χ_Ω before diagonalizing K.

Patient-Specific Basis Functions . . . My Interpretation

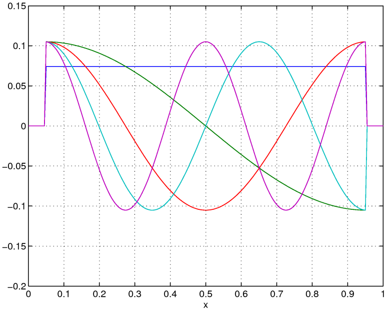
- PSB (Patient-Specific Basis)
  - Pro: The constant DC component vector $\boldsymbol{\chi}_\Omega$ is included.
  - Con 1: *Features near from the boundary of* Ω may not be represented well with a small number of ℓ due to the Dirichlet BC implicitly imposed by $\boldsymbol{\chi}_\Omega$.
  - Con 2: In reality, building a basis for a complicated 3D shape based on the *tensor products* may not be easy, and the boundary effects may become more pronounced.
- LE-CI (Laplacian Eigenfunctions via Commuting Integral Operator)
  - Pro 1: Features near from the boundary may be more efficiently represented thanks to the more natural BC.
  - Pro 2: Building a basis for even a complicated 3D shape is easy; we only need pairwise distances between voxel centers.
  - Con: χ_Ω is not included. However, if we wish, we can include χ_Ω by projecting the kernel matrix K onto the orthogonal complement to span χ_Ω before diagonalizing K.

Patient-Specific Basis Functions . . . My Interpretation

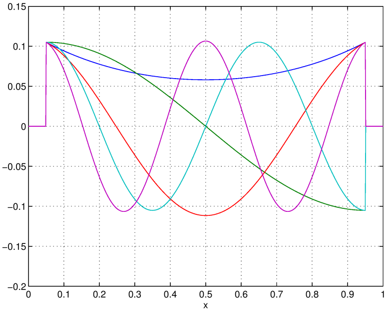
- PSB (Patient-Specific Basis)
  - Pro: The constant DC component vector $\boldsymbol{\chi}_\Omega$ is included.
  - Con 1: *Features near from the boundary of* Ω may not be represented well with a small number of ℓ due to the Dirichlet BC implicitly imposed by $\boldsymbol{\chi}_\Omega$.
  - Con 2: In reality, building a basis for a complicated 3D shape based on the *tensor products* may not be easy, and the boundary effects may become more pronounced.
- LE-CI (Laplacian Eigenfunctions via Commuting Integral Operator)
  - Pro 1: Features near from the boundary may be more efficiently represented thanks to the more natural BC.
  - Pro 2: Building a basis for even a complicated 3D shape is easy; we only need pairwise distances between voxel centers.
  - Con: $\boldsymbol{\chi}_\Omega$ is not included. However, if we wish, we can include $\boldsymbol{\chi}_\Omega$ by projecting the kernel matrix K onto the orthogonal complement to $\operatorname{span} \boldsymbol{\chi}_\Omega$ before diagonalizing K.

Patient-Specific Basis Functions . . . My Interpretation

- *Domain-adapted tensor-product DCT* might be perhaps most computationally efficient without too much boundary effects although 'Con 2' of PSB remains.



(a) Discrete Cosine Basis

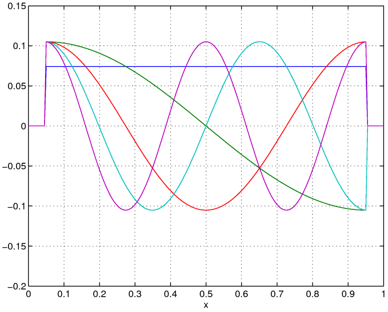


(b) Laplacian Eigenfunctions

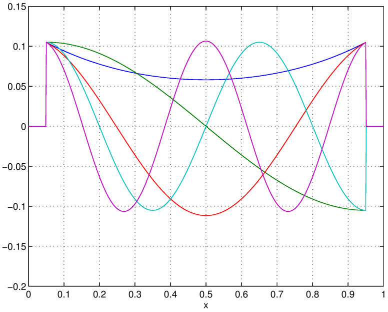
- DCT (Type II) is also used for the *JPEG* image compression standard.
- DCT (Type II) is also the KLT basis for the limiting case of a *1st order Markov process*, i.e., the correlation between adjacent pixels $\rho \uparrow 1$.

Patient-Specific Basis Functions . . . My Interpretation

- *Domain-adapted tensor-product DCT* might be perhaps most computationally efficient without too much boundary effects although 'Con 2' of PSB remains.



(a) Discrete Cosine Basis

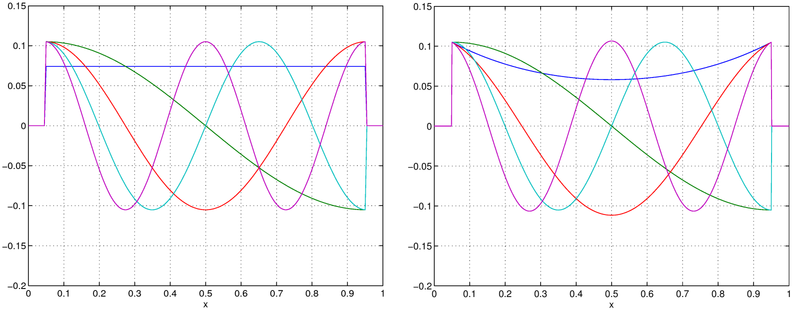


(b) Laplacian Eigenfunctions

- DCT (Type II) is also used for the *JPEG* image compression standard.
- DCT (Type II) is also the KLT basis for the limiting case of a *1st order Markov process*, i.e., the correlation between adjacent pixels $\rho \uparrow 1$.

Patient-Specific Basis Functions . . . My Interpretation

- *Domain-adapted tensor-product DCT* might be perhaps most computationally efficient without too much boundary effects although 'Con 2' of PSB remains.



(a) Discrete Cosine Basis (b) Laplacian Eigenfunctions

- DCT (Type II) is also used for the *JPEG* image compression standard.
- DCT (Type II) is also the KLT basis for the limiting case of a *1st order Markov process*, i.e., the correlation between adjacent pixels $\rho \uparrow 1$.

Outline

Introductory Remarks

- For much more details of this part of lecture, 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.
  - A. E. Brouwer & W. H. Haemers: *Spectra of Graphs*, Springer, 2012.
  - 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.

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

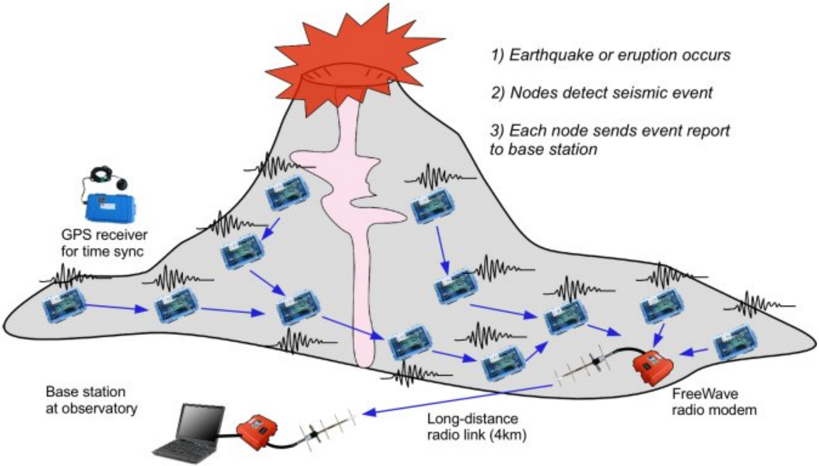


Figure: Volcano monitoring sensor network architecture of Harvard Sensor Networks Lab

An Example of Social Networks

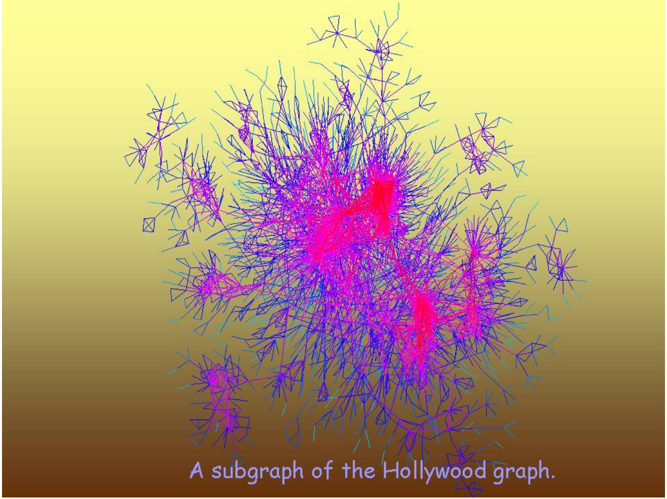


Figure: Through the courtesy of Prof. Fan Chung, UC San Diego

An Example of Biological Networks

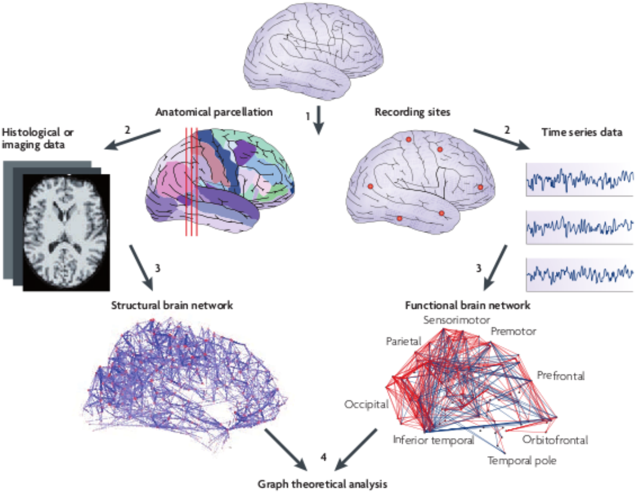
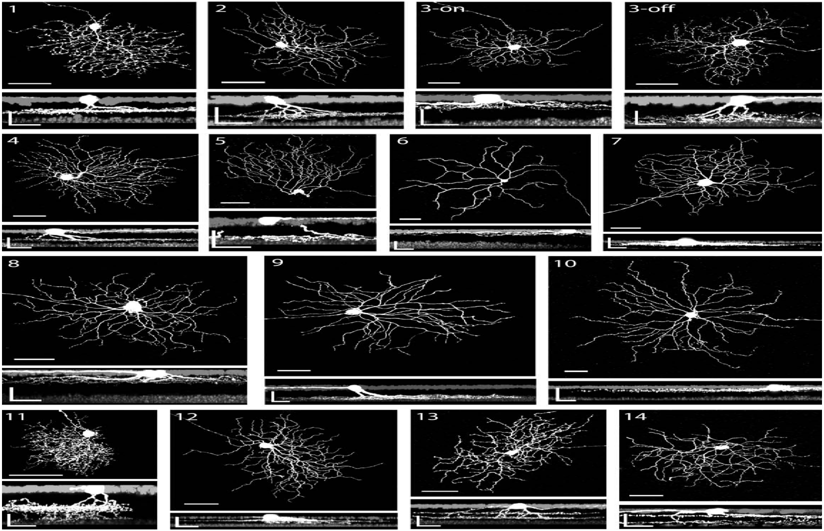
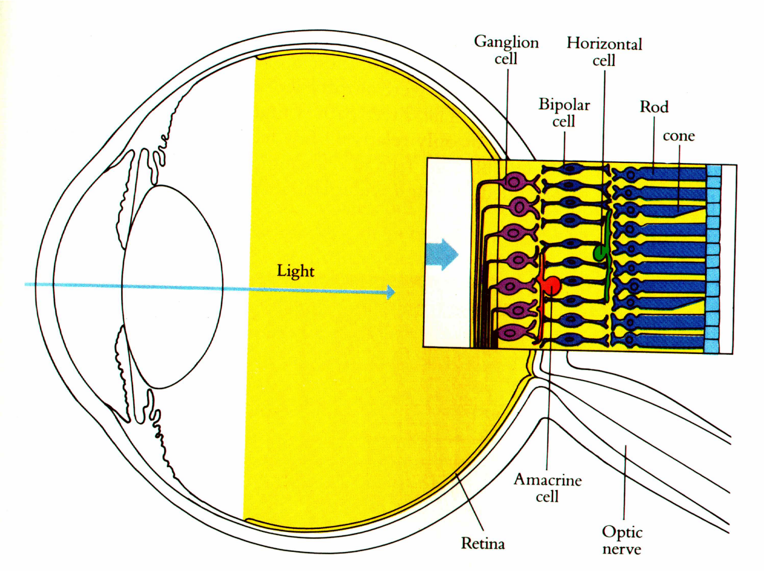


Figure: From E. Bullmore and O. Sporns, *Nature Reviews Neuroscience*, vol. 10, pp.186–198, Mar. 2009.

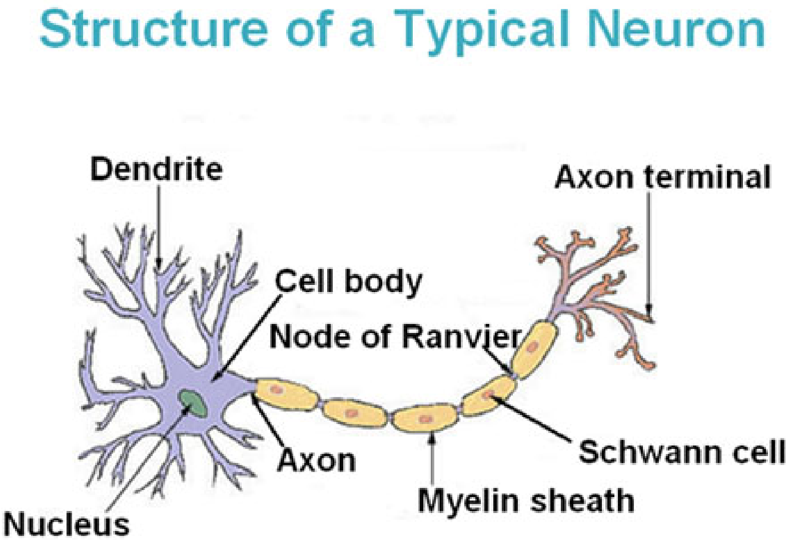
Another Biological Example: Retinal Ganglion Cells



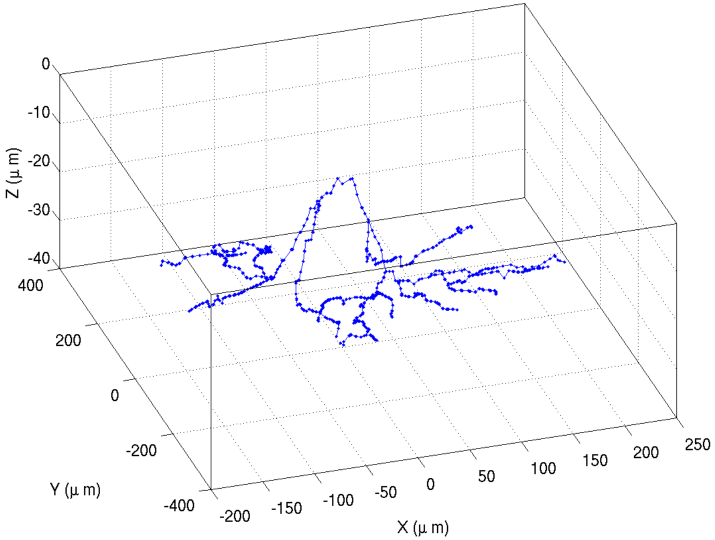
Retinal Ganglion Cells (D. Hubel: *Eye, Brain, & Vision*, '95)



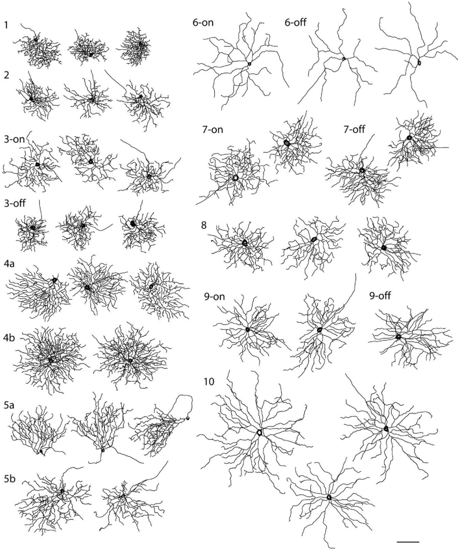
A Typical Neuron (from Wikipedia)



Mouse's RGC as a Graph



Clustering using Features Derived by Neurolucida®



Representing a Regular Image as a Graph

often turns out to be quite useful for various purposes. In particular, Nonlocal Means Denoising Algorithm of Buades-Coll-Morel is quite impressive.

- Construct a graph each of whose vertices represents $k \times k$ patch of a given image (k may be $3, 5, \ldots$, etc.) So each vertex represents a point in $\mathbb{R}^{k^2}$.
- Connect every pair of vertices with the weight $W_{ij} = \exp(-\|\mathrm{patch}_i - \mathrm{patch}_j\|^2/\epsilon^2)$ with *appropriately chosen* scale parameter $\epsilon > 0$.
- Compute the weighted average of the center pixel of each patch using the normalized weights $W_{ij}/\sum_\ell W_{i\ell}$. More precisely, the average of the center of the ith patch, $\overline{c}_i = \sum_j W_{ij} c_j / \sum_\ell W_{i\ell}$.
- See also an interesting work by Daitch-Kelner-Spielman: "Fitting a Graph to Vector Data," *Proc. 26th Intern. Conf. Machine Learning*, 2009.

Representing a Regular Image as a Graph

often turns out to be quite useful for various purposes. In particular, Nonlocal Means Denoising Algorithm of Buades-Coll-Morel is quite impressive.

- Construct a graph each of whose vertices represents $k \times k$ patch of a given image (k may be $3, 5, \ldots$, etc.) So each vertex represents a point in $\mathbb{R}^{k^2}$.
- Connect every pair of vertices with the weight $W_{ij} = \exp(-\|\text{patch}_i - \text{patch}_j\|^2/\epsilon^2)$ with *appropriately chosen* scale parameter $\epsilon > 0$.
- Compute the weighted average of the center pixel of each patch using the normalized weights $W_{ij}/\sum_\ell W_{i\ell}$. More precisely, the average of the center of the ith patch, $\overline{c}_i = \sum_j W_{ij} c_j / \sum_\ell W_{i\ell}$.
- See also an interesting work by Daitch-Kelner-Spielman: "Fitting a Graph to Vector Data," *Proc. 26th Intern. Conf. Machine Learning*, 2009.

Representing a Regular Image as a Graph

often turns out to be quite useful for various purposes. In particular, Nonlocal Means Denoising Algorithm of Buades-Coll-Morel is quite impressive.

- Construct a graph each of whose vertices represents $k \times k$ patch of a given image (k may be $3, 5, \ldots$, etc.) So each vertex represents a point in $\mathbb{R}^{k^2}$.
- Connect every pair of vertices with the weight $W_{ij} = \exp(-\|\text{patch}_i - \text{patch}_j\|^2/\epsilon^2)$ with *appropriately chosen* scale parameter $\epsilon > 0$.
- Compute the weighted average of the center pixel of each patch using the normalized weights $W_{ij}/\sum_\ell W_{i\ell}$. More precisely, the average of the center of the ith patch, $\overline{c}_i = \sum_j W_{ij} c_j / \sum_\ell W_{i\ell}$.
- See also an interesting work by Daitch-Kelner-Spielman: "Fitting a Graph to Vector Data," *Proc. 26th Intern. Conf. Machine Learning*, 2009.

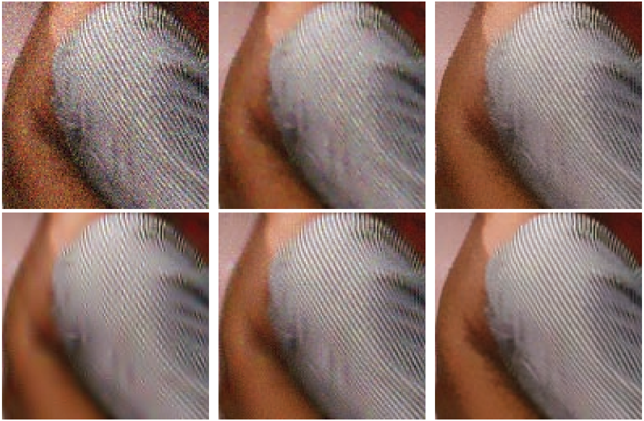
Representing a Regular Image as a Graph

often turns out to be quite useful for various purposes. In particular, Nonlocal Means Denoising Algorithm of Buades-Coll-Morel is quite impressive.

- Construct a graph each of whose vertices represents $k \times k$ patch of a given image (k may be $3, 5, \ldots$, etc.) So each vertex represents a point in $\mathbb{R}^{k^2}$.
- Connect every pair of vertices with the weight $W_{ij} = \exp(-\|\text{patch}_i - \text{patch}_j\|^2/\epsilon^2)$ with *appropriately chosen* scale parameter $\epsilon > 0$.
- Compute the weighted average of the center pixel of each patch using the normalized weights $W_{ij}/\sum_\ell W_{i\ell}$. More precisely, the average of the center of the ith patch, $\overline{c}_i = \sum_j W_{ij} c_j / \sum_\ell W_{i\ell}$.
- See also an interesting work by Daitch-Kelner-Spielman: "Fitting a Graph to Vector Data," *Proc. 26th Intern. Conf. Machine Learning*, 2009.

From: A. Buades, B. Coll, and J.-M. Morel, *SIAM Review*, vol. 52, no. 1, pp. 113–147, 2010.

Noisy Image; Total Variation Denoising; Neighborhood Filter



Trans. Inv. Wavelets; Empirical Wiener; Nonlocal Means

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 lecture, 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 lecture, we shall only deal with simple graphs. So, when we say a graph, we mean a simple graph.

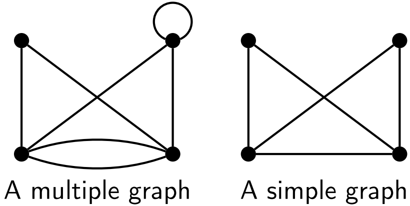
Basic Definitions

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

Basic Definitions

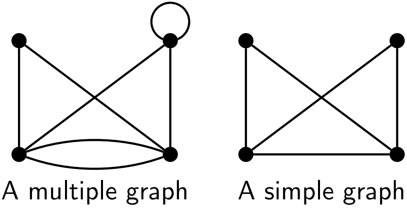
- A graph G consists of a set of vertices (or nodes) V and a set of edges E connecting some pairs of vertices in V. We write $G = (V, E)$.
- An edge connecting a vertex $x \in V$ and itself is called a loop.
- For $x, y \in V$, if $\exists$ more than one edge connecting x and y, they are called multiple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



- In this lecture, 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 lecture, 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 $\boldsymbol{E}$.

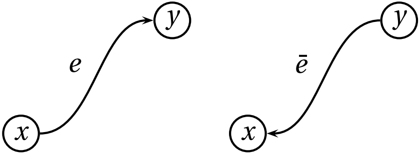
- If $e = [x, y]$, then x and y are called a tail and a head, respectively.

- 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 $\boldsymbol{E}$.
- 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 $\boldsymbol{E}$.

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

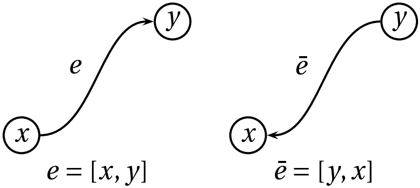
- If two distinct vertices $x, y \in V$ are connected by an edge e, then x, y are called the endpoints (or ends) of e, and x, y are said to be adjacent, and we write $x \sim y$. We also say an edge e is incident with x and y, and e joins x and y.
- The number of edges that are incident with x (i.e., 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 $\boldsymbol{E}$.



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

- 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 $\boldsymbol{E}$.



- 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.
- For a given $x, y \in V$, a sequence of vertices in V $c = (v_1, v_2, \ldots, v_k, v_{k+1})$ is called a path connecting x and y if $v_1 = x$, $v_{k+1} = y$, and $v_1 \sim v_2 \sim \cdots \sim v_k \sim v_{k+1}$. We say the length (or cost) $\ell(c)$ of a path c is the sum of its corresponding edge weights, i.e., $\ell(c) := \sum_{j=1}^{k} w_{v_j, v_{j+1}}$.
- 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.
- $d(x, y) := \inf_{c \in \{\text{paths between } x, y\}} \ell(c)$ is called the graph distance between x and y.
- $\operatorname{diam}(G) := \sup_{x, y \in V} d(x, y)$ is called the diameter of G. Note that $\operatorname{diam}(G) < \infty \Longleftrightarrow G$ is finite.

- 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.
- For a given $x, y \in V$, a sequence of vertices in V $c = (v_1, v_2, \ldots, v_k, v_{k+1})$ is called a path connecting x and y if $v_1 = x$, $v_{k+1} = y$, and $v_1 \sim v_2 \sim \cdots \sim v_k \sim v_{k+1}$. We say the length (or cost) $\ell(c)$ of a path c is the sum of its corresponding edge weights, i.e., $\ell(c) := \sum_{j=1}^{k} w_{v_j, v_{j+1}}$.
- 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.
- $d(x, y) := \inf_{c \in \{\text{paths between } x, y\}} \ell(c)$ is called the graph distance between x and y.
- $\operatorname{diam}(G) := \sup_{x, y \in V} d(x, y)$ is called the diameter of G. Note that $\operatorname{diam}(G) < \infty \iff G$ is finite.

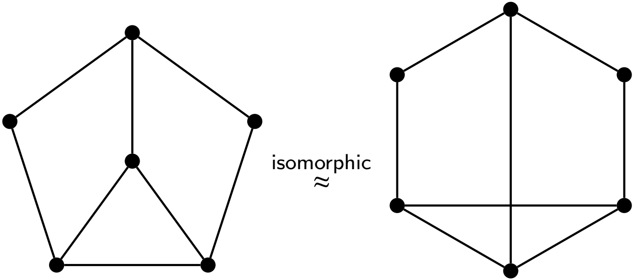
- If an edge e does not have a direction, we write $e = (x, y)$.
- If each edge $e = (x, y)$ of G has a weight (normally positive), written as $w_e = w_{xy}$, then G is called a weighted graph. G is said to be unweighted if $w_e = \text{const.}$ for each $e \in E$, and normally w_e is set to 1.
- For a given $x, y \in V$, a sequence of vertices in V $c = (v_1, v_2, \ldots, v_k, v_{k+1})$ is called a path connecting x and y if $v_1 = x$, $v_{k+1} = y$, and $v_1 \sim v_2 \sim \cdots \sim v_k \sim v_{k+1}$. We say the length (or cost) $\ell(c)$ of a path c is the sum of its corresponding edge weights, i.e., $\ell(c) := \sum_{j=1}^{k} w_{v_j, v_{j+1}}$.
- 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.
- $d(x, y) := \inf_{c \in \{\text{paths between } x, y\}} \ell(c)$ is called the graph distance between x and y.
- $\operatorname{diam}(G) := \sup_{x,y \in V} d(x, y)$ is called the diameter of G. Note that $\operatorname{diam}(G) < \infty \iff G$ is finite.

- 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.
- For a given $x, y \in V$, a sequence of vertices in V $c = (v_1, v_2, \dots, v_k, v_{k+1})$ is called a path connecting x and y if $v_1 = x$, $v_{k+1} = y$, and $v_1 \sim v_2 \sim \cdots \sim v_k \sim v_{k+1}$. We say the length (or cost) $\ell(c)$ of a path c is the sum of its corresponding edge weights, i.e., $\ell(c) := \sum_{j=1}^{k} w_{v_j, v_{j+1}}$.
- 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.
- $d(x,y) := \inf_{c \in \{\text{paths between } x, y\}} \ell(c)$ is called the graph distance between x and y.
- $\operatorname{diam}(G) := \sup_{x,y \in V} d(x,y)$ is called the diameter of G. Note that $\operatorname{diam}(G) < \infty \iff G$ is finite.

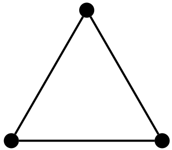
- If an edge e does not have a direction, we write $e = (x, y)$.
- If each edge $e = (x, y)$ of G has a weight (normally positive), written as $w_e = w_{xy}$, then G is called a weighted graph. G is said to be unweighted if $w_e = \text{const.}$ for each $e \in E$, and normally w_e is set to 1.
- For a given $x, y \in V$, a sequence of vertices in V $c = (v_1, v_2, \ldots, v_k, v_{k+1})$ is called a path connecting x and y if $v_1 = x$, $v_{k+1} = y$, and $v_1 \sim v_2 \sim \cdots \sim v_k \sim v_{k+1}$. We say the length (or cost) $\ell(c)$ of a path c is the sum of its corresponding edge weights, i.e., $\ell(c) := \sum_{j=1}^{k} w_{v_j, v_{j+1}}$.
- 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.
- $d(x, y) := \inf_{c \in \{\text{paths between } x, y\}} \ell(c)$ is called the graph distance between x and y.
- $\operatorname{diam}(G) := \sup_{x,y \in V} d(x, y)$ is called the diameter of G. Note that $\operatorname{diam}(G) < \infty \iff G$ is finite.

- 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.
- For a given $x, y \in V$, a sequence of vertices in V $c = (v_1, v_2, \ldots, v_k, v_{k+1})$ is called a path connecting x and y if $v_1 = x$, $v_{k+1} = y$, and $v_1 \sim v_2 \sim \cdots \sim v_k \sim v_{k+1}$. We say the length (or cost) $\ell(c)$ of a path c is the sum of its corresponding edge weights, i.e., $\ell(c) := \sum_{j=1}^{k} w_{v_j, v_{j+1}}$.
- 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.
- $d(x, y) := \inf_{c \in \{\text{paths between } x, y\}} \ell(c)$ is called the graph distance between x and y.
- $\operatorname{diam}(G) := \sup_{x, y \in V} d(x, y)$ is called the diameter of G. Note that $\operatorname{diam}(G) < \infty \iff G$ is finite.

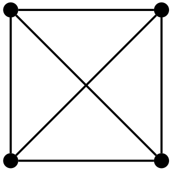
- We say two graphs are isomorphic if $\exists$ a one-to-one correspondence between the vertex sets such that if two vertices are joined by an edge in one graph, the corresponding vertices are also joined by an edge in the other graph.



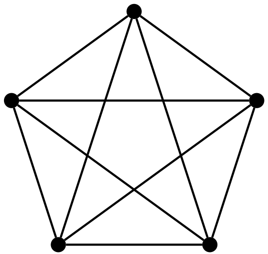
- The complete graph K_n on n vertices is a simple graph that has all possible $\binom{n}{2}$ edges.



K_3



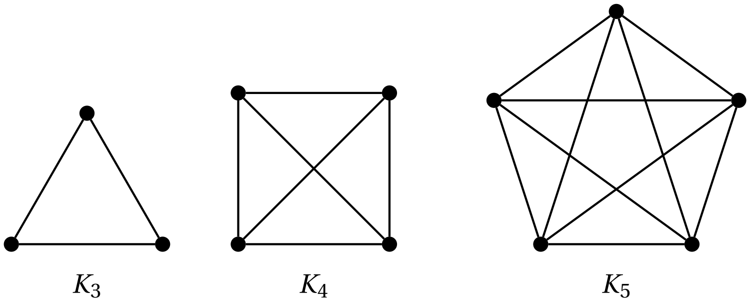
K_4



K_5

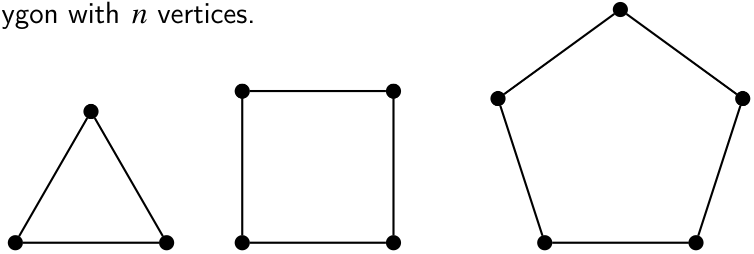
- If all the vertices of a graph has the same degree, the graph is called regular. Hence, K_n is regular.

- 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.

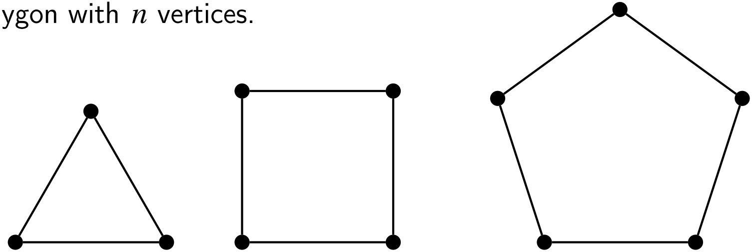
- A polygon is a finite connected graph that is regular of degree 2. $P_n =$ a polygon with n vertices.



$P_3 = K_3$ $\quad$ P_4 $\quad$ P_5

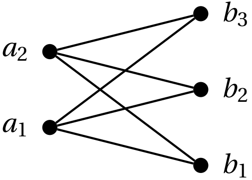
- The complete bipartite graph $K_{n,m}$ has $n+m$ vertices $a_1, \ldots, a_n$, $b_1, \ldots, b_m$, and all nm pairs (a_i, b_j) as edges. An example: $K_{2,3}$:

- A polygon is a finite connected graph that is regular of degree 2. $P_n =$ a polygon with n vertices.



$P_3 = K_3$ P_4 P_5

- The complete bipartite graph $K_{n,m}$ has $n+m$ vertices $a_1, \ldots, a_n$, $b_1, \ldots, b_m$, and all nm pairs (a_i, b_j) as edges. An example: $K_{2,3}$:



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 $D = D(G) = \text{diag}(d_1, \ldots, d_n) \in \mathbb{R}^{n \times n}$ 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 . . .

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

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

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

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

- The signless Laplacian is defined as follows, but we will not deal with this in this lecture: $Q(G) := D + A$.
- Graph Laplacians can also be defined for directed graphs; see, e.g., Fan Chung: "Laplacians and the Cheeger inequality for directed graphs," *Ann. Comb.*, vol. 9, no. 1, pp. 1–19, 2005.

Matrices Associated with a Graph . . .

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

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

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

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

- The signless Laplacian is defined as follows, but we will not deal with this in this lecture: $Q(G) := D + A$.
- Graph Laplacians can also be defined for directed graphs; see, e.g., Fan Chung: "Laplacians and the Cheeger inequality for directed graphs," *Ann. Comb.*, vol. 9, no. 1, pp. 1–19, 2005.

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 lecture: $Q(G) := D + A$.
- Graph Laplacians can also be defined for directed graphs; see, e.g., Fan Chung: "Laplacians and the Cheeger inequality for directed graphs," *Ann. Comb.*, vol. 9, no. 1, pp. 1–19, 2005.

Functions Defined on a Graph

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

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

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

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

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

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

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

- These corresponds to:

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

- 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 $C(\boldsymbol{E}) := \{\varphi \text{ defined on } \boldsymbol{E} \,|\, \varphi(\bar{e}) = -\varphi(e), e \in \boldsymbol{E}\}$. 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)

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

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

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

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

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

Corollary

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

$$\langle L_{\mathrm{rw}} f, f \rangle = \sum_{u \in V} L f(u) f(u) = \langle df, df \rangle \geq 0.$$

The Minimum Principle

Theorem (The discrete version of the minimum principle)

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

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

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

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

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

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 lecture, 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 lecture, 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 lecture, 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 lecture, 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 lecture, 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 lecture, 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 lecture, 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 $0 = \lambda_0(G) \leq \lambda_1(G) \leq \cdots \leq \lambda_{n-1}(G)$ 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 P such that
$$L(G_2) = P^{\mathsf{T}} L(G_1) P.$$
- $\operatorname{rank} L(G) = n - m_G(0)$ 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 P such that
$$L(G_2) = P^{\mathsf{T}} L(G_1) P.$$
- $\operatorname{rank} L(G) = n - m_G(0)$ 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 . . .

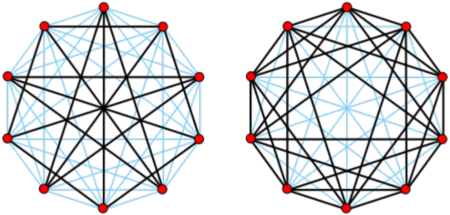
- Graph Laplacian matrices of the same graph are permutation-similar. In fact, graphs G_1 and G_2 are *isomorphic* iff there exists a permutation matrix P such that
$$L(G_2) = P^{\mathsf{T}} L(G_1) P.$$
- $\operatorname{rank} L(G) = n - m_G(0)$ where $m_G(0)$ turns out to be the number of connected components 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 P such that
$$L(G_2) = P^{\top} L(G_1) P.$$
- $\operatorname{rank} L(G) = n - m_G(0)$ 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

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

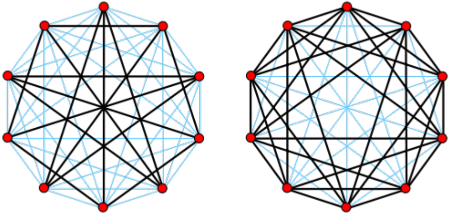
where J_n is the $n \times n$ matrix whose entries are all 1.

- 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

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

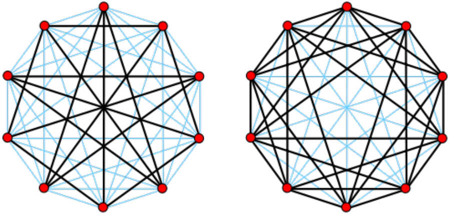
where J_n is the $n \times n$ matrix whose entries are all 1.

- 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

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

A Brief Review of Graph Laplacian Eigenvalues . . .

- From the above, we can see that

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

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

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

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

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

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 $\lambda_j > 0$, $j = 1, \ldots, n-1$, 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 $\phi(x) = 0$, 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 $n-1$.

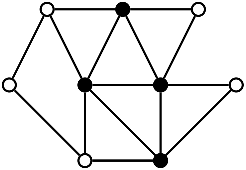
Proof. Let $v \in V$ be a vertex with $d(v) = n-1$. Then, $L\boldsymbol{\phi}(v) = (n-1)\boldsymbol{\phi}(v) - \sum_{u \neq v} \boldsymbol{\phi}(u) = \lambda \boldsymbol{\phi}(v)$. But, $\boldsymbol{\phi} \perp \mathbf{1}_n$, so $\sum_{u \neq v} \boldsymbol{\phi}(u) = -\boldsymbol{\phi}(v)$. This leads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

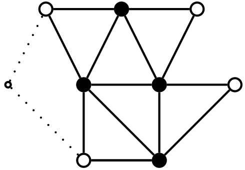
Let W be a nonempty subset of $V(G)$. Then, the reduced graph $G\{W\}$ is obtained from G by deleting all vertices in $V \setminus W$ that are not adjacent to a vertex of W and subsequent deletion of any remaining edges that are not incident with a vertex in W.



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

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 ...

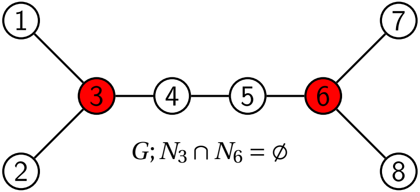
Theorem (Merris (1998))

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))

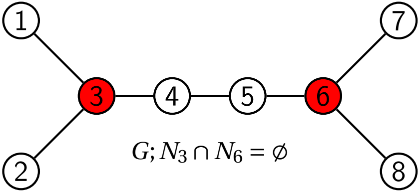
Let $\boldsymbol{\phi}$ be an eigenfunction affording λ of G. Let N_v be the set of neighbors of v. Suppose $\boldsymbol{\phi}(u) = \boldsymbol{\phi}(v) = 0$, where $N_u \cap N_v = \emptyset$. Let G' be the graph on $n-1$ vertices obtained by coalescing u and v into a single vertex, which is adjacent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

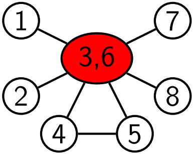


$$\lambda_2(G) = 1; \boldsymbol{\phi}_2(G) = [-0.0261, -0.0261, 0, 0.0523, 0.0523, 0, -0.7303, 0.6781]^{\mathsf{T}}$$

A Simple Example



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



G'

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

Outline

The Perron-Frobenius Theorem

Let $A \in \mathbb{R}^{n \times n}$ be a rather general symmetric matrix associated with a graph G such that $A_{uv} \neq 0$ iff $e = (u, v) \in E(G)$. Then, A is called irreducible if its underlying graph is connected.

Theorem (Perron-Frobenius Theorem)

Let A, B be real symmetric irreducible nonnegative $n \times n$ matrices. Then,

(i) *the spectral radius $\rho(A)$ is a simple eigenvalue of A. If ϕ is an eigenfunction for $\rho(A)$, then no entries of ϕ are zero, and all have the same sign.*

(ii) *Furthermore, if $A - B$ is nonnegative, then $\rho(B) \leq \rho(A)$, with equality iff $B = A$.*

Corollary

Let G be a connected graph. Then, the smallest eigenvalue of $L(G)$, $L_{\mathrm{rw}}(G)$, $L_{\mathrm{sym}}(G)$, i.e., $\lambda_0 = 0$, is simple, and ϕ_0 can be taken to have all entries positive. ϕ_0 is often called the Perron vector of G.

The Perron-Frobenius Theorem

Let $A \in \mathbb{R}^{n\times n}$ be a rather general symmetric matrix associated with a graph G such that $A_{uv} \neq 0$ iff $e = (u, v) \in E(G)$. Then, A is called irreducible if its underlying graph is connected.

Theorem (Perron-Frobenius Theorem)

Let A, B be real symmetric irreducible nonnegative $n \times n$ matrices. Then,

(i) *the spectral radius $\rho(A)$ is a simple eigenvalue of A. If $\boldsymbol{\phi}$ is an eigenfunction for $\rho(A)$, then no entries of $\boldsymbol{\phi}$ are zero, and all have the same sign.*

(ii) *Furthermore, if $A - B$ is nonnegative, then $\rho(B) \leq \rho(A)$, with equality iff $B = A$.*

Corollary

Let G be a connected graph. Then, the smallest eigenvalue of $L(G)$, $L_{\mathrm{rw}}(G)$, $L_{\mathrm{sym}}(G)$, i.e., $\lambda_0 = 0$, is simple, and $\boldsymbol{\phi}_0$ can be taken to have all entries positive. $\boldsymbol{\phi}_0$ is often called the Perron vector of G.

The Perron-Frobenius Theorem

Let $A \in \mathbb{R}^{n \times n}$ be a rather general symmetric matrix associated with a graph G such that $A_{uv} \neq 0$ iff $e = (u, v) \in E(G)$. Then, A is called irreducible if its underlying graph is connected.

Theorem (Perron-Frobenius Theorem)

Let A, B be real symmetric irreducible nonnegative $n \times n$ matrices. Then,

(i) *the spectral radius $\rho(A)$ is a simple eigenvalue of A. If $\boldsymbol{\phi}$ is an eigenfunction for $\rho(A)$, then no entries of $\boldsymbol{\phi}$ are zero, and all have the same sign.*

(ii) *Furthermore, if $A - B$ is nonnegative, then $\rho(B) \leq \rho(A)$, with equality iff $B = A$.*

Corollary

Let G be a connected graph. Then, the smallest eigenvalue of $L(G)$, $L_{\mathrm{rw}}(G)$, $L_{\mathrm{sym}}(G)$, i.e., $\lambda_0 = 0$, is simple, and $\boldsymbol{\phi}_0$ can be taken to have all entries positive. $\boldsymbol{\phi}_0$ is often called the Perron vector of G.

My Comments on the Perron-Frobenius Theorem

- If $G = P_n$, then $\boldsymbol{\phi}_j$ is jth DCT-II basis vector, as I discussed before. Hence, the Perron vector of P_n is the constant vector for the DC component in the signal processing terminology.
- For the continuous case, I talked about the integral operator $\mathcal{K}$ that commutes with the Laplace operator. In particular, I showed the 1D example where the domain is the unit interval $\Omega = (0,1)$. In that case, the smallest eigenvalue is $\lambda_0 \approx -5.756915$, and $\phi_0(x) \propto \cosh \sqrt{-\lambda_0}\left(x - \frac{1}{2}\right)$. This function also does not change its sign, hence it can be viewed as the Perron vector of $\mathcal{K}$.

My Comments on the Perron-Frobenius Theorem

- If $G = P_n$, then $\boldsymbol{\phi}_j$ is jth DCT-II basis vector, as I discussed before. Hence, the Perron vector of P_n is the constant vector for the DC component in the signal processing terminology.
- For the continuous case, I talked about the integral operator $\mathscr{K}$ that commutes with the Laplace operator. In particular, I showed the 1D example where the domain is the unit interval $\Omega = (0,1)$. In that case, the smallest eigenvalue is $\lambda_0 \approx -5.756915$, and $\phi_0(x) \propto \cosh\sqrt{-\lambda_0}\left(x - \frac{1}{2}\right)$. This function also does not change its sign, hence it can be viewed as the Perron vector of $\mathscr{K}$.

My Comments on the Perron-Frobenius Theorem . . .

- Does there exist the P-F theory for compact operators? $\Longrightarrow$ YES!

Theorem (Krein & Rutman (1948))

Let X be a Banach space, and let $K \subset X$ be a convex cone such that the set $K - K = \{f - g \mid f, g \in K\}$ is dense in X. Let $T : X \to X$ be a non-zero compact operator which is positive, meaning that $T(K) \subset K$, and assume that its spectral radius $\rho(T)$ is strictly positive. Then $\rho(T)$ is an eigenvalue of T with positive eigenfunction, meaning that there exists $\phi \in K \setminus \{0\}$ such that $T(\phi) = \rho(T)\phi$.

- Generally, one of my research goals is to consider *the graph version of the integral operator commuting with a given graph Laplacian*, and analyze its properties!

My Comments on the Perron-Frobenius Theorem . . .

- Does there exist the P-F theory for compact operators? $\Longrightarrow$ YES!

Theorem (Krein & Rutman (1948))

Let X be a Banach space, and let $K \subset X$ be a convex cone such that the set $K - K = \{f - g \mid f, g \in K\}$ is dense in X. Let $T : X \to X$ be a non-zero compact operator which is positive, meaning that $T(K) \subset K$, and assume that its spectral radius $\rho(T)$ is strictly positive. Then $\rho(T)$ is an eigenvalue of T with positive eigenfunction, meaning that there exists $\phi \in K \setminus \{0\}$ such that $T(\phi) = \rho(T)\phi$.

- Generally, one of my research goals is to consider *the graph version of the integral operator commuting with a given graph Laplacian*, and analyze its properties!

My Comments on the Perron-Frobenius Theorem . . .

- Does there exist the P-F theory for compact operators? $\Longrightarrow$ YES!

Theorem (Krein & Rutman (1948))

Let X be a Banach space, and let $K \subset X$ be a convex cone such that the set $K - K = \{f - g \mid f, g \in K\}$ is dense in X. Let $T : X \to X$ be a non-zero compact operator which is positive, meaning that $T(K) \subset K$, and assume that its spectral radius $\rho(T)$ is strictly positive. Then $\rho(T)$ is an eigenvalue of T with positive eigenfunction, meaning that there exists $\phi \in K \setminus \{0\}$ such that $T(\phi) = \rho(T)\phi$.

- Generally, one of my research goals is to consider *the graph version of the integral operator commuting with a given graph Laplacian*, and analyze its properties!

Outline

Perron-Frobenius/Fiedler $\Longrightarrow$ Courant

- From the Perron-Frobenius theorem, for a connected graph G, we know that $\lambda_0 = 0$ and $\boldsymbol{\phi}_0$ is all positive.
- By Fielder, we also know that the algebraic connectivity $a(G) = \lambda_1(G) > 0$, $\boldsymbol{\phi}_1$ (called the Fiedler vector of G) splits V into three subsets $V = V_+ \cup V_- \cup V_0$ where the values of $\boldsymbol{\phi}_1$ on V_+, V_-, V_0 are positive, negative, and zero (note that V_0 could be $\emptyset$).
- These reminds us of Courant's celebrated Nodal Domain Theorem for elliptic operatros on manifolds.

Perron-Frobenius/Fiedler $\Longrightarrow$ Courant

- From the Perron-Frobenius theorem, for a connected graph G, we know that $\lambda_0 = 0$ and $\boldsymbol{\phi}_0$ is all positive.
- By Fielder, we also know that the algebraic connectivity $a(G) = \lambda_1(G) > 0$, $\boldsymbol{\phi}_1$ (called the Fiedler vector of G) splits V into three subsets $V = V_+ \cup V_- \cup V_0$ where the values of $\boldsymbol{\phi}_1$ on V_+, V_-, V_0 are positive, negative, and zero (note that V_0 could be $\emptyset$).
- These reminds us of Courant's celebrated Nodal Domain Theorem for elliptic operatros on manifolds.

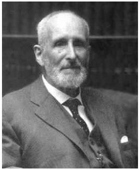
Perron-Frobenius/Fiedler $\Longrightarrow$ Courant

- From the Perron-Frobenius theorem, for a connected graph G, we know that $\lambda_0 = 0$ and $\boldsymbol{\phi}_0$ is all positive.
- By Fielder, we also know that the algebraic connectivity $a(G) = \lambda_1(G) > 0$, $\boldsymbol{\phi}_1$ (called the Fiedler vector of G) splits V into three subsets $V = V_+ \cup V_- \cup V_0$ where the values of $\boldsymbol{\phi}_1$ on V_+, V_-, V_0 are positive, negative, and zero (note that V_0 could be $\emptyset$).
- These reminds us of Courant's celebrated Nodal Domain Theorem for elliptic operatros on manifolds.

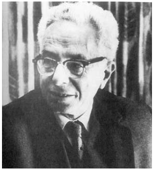
Perron-Frobenius/Fiedler $\Longrightarrow$ Courant

- From the Perron-Frobenius theorem, for a connected graph G, we know that $\lambda_0 = 0$ and $\boldsymbol{\phi}_0$ is all positive.
- By Fielder, we also know that the algebraic connectivity $a(G) = \lambda_1(G) > 0$, $\boldsymbol{\phi}_1$ (called the Fiedler vector of G) splits V into three subsets $V = V_+ \cup V_- \cup V_0$ where the values of $\boldsymbol{\phi}_1$ on V_+, V_-, V_0 are positive, negative, and zero (note that V_0 could be $\emptyset$).
- These reminds us of Courant's celebrated Nodal Domain Theorem for elliptic operatros on manifolds.

(a) F. G. Frobenius (1849–1917)



(b) Oskar Perron (1880–1975)



(c) Richard Courant (1888–1972)

(d) Miroslav Fiedler (1926–)

Courant's Nodal Domain Theorem

Theorem (Courant (1923))

Let $\mathscr{L}$ be a self-adjoint second order differential operator, and consider the following elliptic eigenvalue problem on a domain $\Omega \subset \mathbb{R}^d$:

$$\mathscr{L}u + \lambda \rho u = 0, \quad \rho > 0,$$

with arbitrary homogeneous boundary conditions. If its eigenfunctions are ordered according to increasing eigenvalues, then the nodes (a.k.a. nodal sets or nodal lines) of the kth eigenfunction ϕ_k $(k = 0, 1, \dots)$ divide Ω into no more than $k+1$ subdomains.

Of course, the nodal sets of a function $f(x)$ in Ω is defined as

$$\mathfrak{N}[f] := \{x \in \Omega \mid f(x) = 0\}.$$

Courant's Nodal Domain Theorem

Theorem (Courant (1923))

Let $\mathscr{L}$ be a self-adjoint second order differential operator, and consider the following elliptic eigenvalue problem on a domain $\Omega \subset \mathbb{R}^d$:

$$\mathscr{L}u + \lambda \rho u = 0, \quad \rho > 0,$$

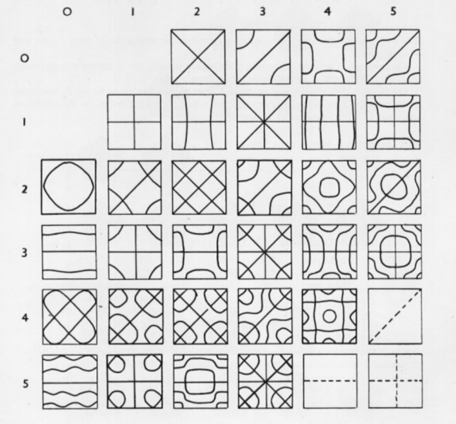
with arbitrary homogeneous boundary conditions. If its eigenfunctions are ordered according to increasing eigenvalues, then the nodes (a.k.a. nodal sets or nodal lines) of the kth eigenfunction ϕ_k $(k = 0, 1, \ldots)$ divide Ω into no more than $k+1$ subdomains.

Of course, the nodal sets of a function $f(\boldsymbol{x})$ in Ω is defined as

$$\mathfrak{N}[f] := \{\boldsymbol{x} \in \Omega \mid f(\boldsymbol{x}) = 0\}.$$

A Famous Example of Nodal Domain Theorem

Courtesy: `http://www.cymascope.com/cyma_research/history.html`



(a) Chladni Plates

(b) Ernst Chladini (1756–1827)

Discrete Nodal Domains

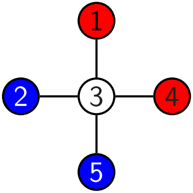
- In the context of manifolds, the nodal domains of f refers to the connected components of the complement of the nodal set $\mathfrak{N}[f]$, i.e., to the components of $\{\boldsymbol{x} \in \Omega \mid f(\boldsymbol{x}) \neq 0\}$, which are bounded by the nodal sets.
- The discrete analog of a "nodal domain" is a maximal connected induced subgraph consisting entirely of positive and negative vertices w.r.t. a given function f defined over $V(G)$.
- However, more subtlety comes in:

Discrete Nodal Domains

- In the context of manifolds, the nodal domains of f refers to the connected components of the complement of the nodal set $\mathfrak{N}[f]$, i.e., to the components of $\{\boldsymbol{x} \in \Omega \mid f(\boldsymbol{x}) \neq 0\}$, which are bounded by the nodal sets.
- The discrete analog of a "nodal domain" is a maximal connected induced subgraph consisting entirely of positive and negative vertices w.r.t. a given function f defined over $V(G)$.
- However, more subtlety comes in:

Discrete Nodal Domains

- In the context of manifolds, the nodal domains of f refers to the connected components of the complement of the nodal set $\mathfrak{N}[f]$, i.e., to the components of $\{\boldsymbol{x} \in \Omega \,|\, f(\boldsymbol{x}) \neq 0\}$, which are bounded by the nodal sets.
- The discrete analog of a "nodal domain" is a maximal connected induced subgraph consisting entirely of positive and negative vertices w.r.t. a given function f defined over $V(G)$.
- However, more subtlety comes in:



$K_{1,4}$

$$\lambda_1 = 1;\ m_{K_{1,4}}(1) = 3;\ \boldsymbol{\phi}_1 \propto [1, -1, 0, 1, -1]^{\mathsf{T}}.$$

Discrete Nodal Domains ...

- A positive (or negative) strong nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) > 0$ (or $f(v) < 0$). The number of strong nodal domains of f is denoted by $\mathfrak{S}(f)$.
- In contrast, a positive (or negative) weak nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) \geq 0$ (or $f(v) \leq 0$) that contains at least one nonzero vertex. The number of weak nodal domains of f is denoted by $\mathfrak{W}(f)$.
- In the above example of $K_{1,4}$, $\mathfrak{S}(\phi_1) = 4$ and $\mathfrak{W}(\phi_1) = 2$ because the strong nodal domains are $\{\{1\},\{2\},\{4\},\{5\}\}$ while the weak nodal domains are $\{\{1,3,4\},\{2,3,5\}\}$.
- Obviously, we always have $\mathfrak{W}(f) \leq \mathfrak{S}(f)$.
- The zero vertices separate positive (or negative) strong nodal domains while they join weak nodal domains. In fact, each zero vertex simultaneously belongs to exactly one weak positive nodal domain and exactly one weak negative nodal domain.

Discrete Nodal Domains . . .

- A positive (or negative) strong nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) > 0$ (or $f(v) < 0$). The number of strong nodal domains of f is denoted by $\mathfrak{S}(f)$.
- In contrast, a positive (or negative) weak nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) \geq 0$ (or $f(v) \leq 0$) that contains at least one nonzero vertex. The number of weak nodal domains of f is denoted by $\mathfrak{W}(f)$.
- In the above example of $K_{1,4}$, $\mathfrak{S}(\phi_1) = 4$ and $\mathfrak{W}(\phi_1) = 2$ because the strong nodal domains are $\{\{1\},\{2\},\{4\},\{5\}\}$ while the weak nodal domains are $\{\{1,3,4\},\{2,3,5\}\}$.
- Obviously, we always have $\mathfrak{W}(f) \leq \mathfrak{S}(f)$.
- The zero vertices separate positive (or negative) strong nodal domains while they join weak nodal domains. In fact, each zero vertex simultaneously belongs to exactly one weak positive nodal domain and exactly one weak negative nodal domain.

Discrete Nodal Domains ...

- A positive (or negative) strong nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) > 0$ (or $f(v) < 0$). The number of strong nodal domains of f is denoted by $\mathfrak{S}(f)$.
- In contrast, a positive (or negative) weak nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) \geq 0$ (or $f(v) \leq 0$) that contains at least one nonzero vertex. The number of weak nodal domains of f is denoted by $\mathfrak{W}(f)$.
- In the above example of $K_{1,4}$, $\mathfrak{S}(\boldsymbol{\phi}_1) = 4$ and $\mathfrak{W}(\boldsymbol{\phi}_1) = 2$ because the strong nodal domains are $\{\{1\},\{2\},\{4\},\{5\}\}$ while the weak nodal domains are $\{\{1,3,4\},\{2,3,5\}\}$.
- Obviously, we always have $\mathfrak{W}(f) \leq \mathfrak{S}(f)$.
- The zero vertices separate positive (or negative) strong nodal domains while they join weak nodal domains. In fact, each zero vertex simultaneously belongs to exactly one weak positive nodal domain and exactly one weak negative nodal domain.

Discrete Nodal Domains ...

- A positive (or negative) strong nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) > 0$ (or $f(v) < 0$). The number of strong nodal domains of f is denoted by $\mathfrak{S}(f)$.
- In contrast, a positive (or negative) weak nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) \geq 0$ (or $f(v) \leq 0$) that contains at least one nonzero vertex. The number of weak nodal domains of f is denoted by $\mathfrak{W}(f)$.
- In the above example of $K_{1,4}$, $\mathfrak{S}(\boldsymbol{\phi}_1) = 4$ and $\mathfrak{W}(\boldsymbol{\phi}_1) = 2$ because the strong nodal domains are $\{\{1\},\{2\},\{4\},\{5\}\}$ while the weak nodal domains are $\{\{1,3,4\},\{2,3,5\}\}$.
- Obviously, we always have $\mathfrak{W}(f) \leq \mathfrak{S}(f)$.
- The zero vertices separate positive (or negative) strong nodal domains while they join weak nodal domains. In fact, each zero vertex simultaneously belongs to exactly one weak positive nodal domain and exactly one weak negative nodal domain.

Discrete Nodal Domains ...

- A positive (or negative) strong nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) > 0$ (or $f(v) < 0$). The number of strong nodal domains of f is denoted by $\mathfrak{S}(f)$.
- In contrast, a positive (or negative) weak nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) \geq 0$ (or $f(v) \leq 0$) that contains at least one nonzero vertex. The number of weak nodal domains of f is denoted by $\mathfrak{W}(f)$.
- In the above example of $K_{1,4}$, $\mathfrak{S}(\boldsymbol{\phi}_1) = 4$ and $\mathfrak{W}(\boldsymbol{\phi}_1) = 2$ because the strong nodal domains are $\{\{1\},\{2\},\{4\},\{5\}\}$ while the weak nodal domains are $\{\{1,3,4\},\{2,3,5\}\}$.
- Obviously, we always have $\mathfrak{W}(f) \leq \mathfrak{S}(f)$.
- The zero vertices separate positive (or negative) strong nodal domains while they join weak nodal domains. In fact, each zero vertex simultaneously belongs to exactly one weak positive nodal domain and exactly one weak negative nodal domain.

Discrete Nodal Domains . . .

We focus our attention on the kth eigenvalue λ_k with multiplicity r of a graph Laplacian $(L,\ L_{\mathrm{rw}},\ L_{\mathrm{sym}})$.

$$\lambda_0 \leq \lambda_1 \leq \cdots \lambda_{k-1} < \lambda_k = \lambda_{k+1} = \cdots = \lambda_{k+r-1} < \lambda_{k+r} \leq \cdots \leq \lambda_{n-1}.$$

Discrete Nodal Domains . . .

We focus our attention on the kth eigenvalue λ_k with multiplicity r of a graph Laplacian (L, L_{rw}, L_{sym}).

$$\lambda_0 \leq \lambda_1 \leq \cdots \lambda_{k-1} < \lambda_k = \lambda_{k+1} = \cdots = \lambda_{k+r-1} < \lambda_{k+r} \leq \cdots \leq \lambda_{n-1}.$$

Theorem (Discrete Nordal Domain Theorem (Davies, Gladwell, Leydold, Stadler, 2001))

Let G be a connected graph with n vertices. Then, any graph Laplacian eigenfunction $\boldsymbol{\phi}_k$ corresponding to λ_k with multiplicity r has at most $k+1$ weak nodal domains and $k+r$ strong nodal domains, i.e.,

$$\mathfrak{W}(\boldsymbol{\phi}_k) \leq k+1, \quad \mathfrak{S}(\boldsymbol{\phi}_k) \leq k+r$$

where $k \in [0, n-1]$.

Discrete Nodal Domains ...

We focus our attention on the kth eigenvalue λ_k with multiplicity r of a graph Laplacian (L, L_{rw}, L_{sym}).

$$\lambda_0 \le \lambda_1 \le \cdots \lambda_{k-1} < \lambda_k = \lambda_{k+1} = \cdots = \lambda_{k+r-1} < \lambda_{k+r} \le \cdots \le \lambda_{n-1}.$$

Theorem (Discrete Nordal Domain Theorem (Davies, Gladwell, Leydold, Stadler, 2001))

Let G be a connected graph with n vertices. Then, any graph Laplacian eigenfunction $\boldsymbol{\phi}_k$ corresponding to λ_k with multiplicity r has at most $k+1$ weak nodal domains and $k+r$ strong nodal domains, i.e.,

$$\mathfrak{W}(\boldsymbol{\phi}_k) \le k+1, \quad \mathfrak{S}(\boldsymbol{\phi}_k) \le k+r$$

where $k \in [0, n-1]$.

In the example of $K_{1,4}$, $\lambda_1 = 1$ has multiplicity $r = 3$. Hence, $\mathfrak{W}(\boldsymbol{\phi}_1) = 2 \le 1+1$ and $\mathfrak{S}(\boldsymbol{\phi}_1) = 4 \le 1+3$ are satisfied!

Discrete Nodal Domains . . .

Corollary (Fiedler (1975))

If G is connected, then $\mathfrak{W}(\boldsymbol{\phi}_1) = 2$.

Corollary (Fiedler (1975))

The eigenfunction $\boldsymbol{\phi}_k$ affording λ_k has at most k positive weak nodal domains for $k \geq 1$. Consequently, $\mathfrak{W}(\boldsymbol{\phi}_k) \leq 2k$.

Discrete Nodal Domains . . .

Corollary (Fiedler (1975))

If G is connected, then $\mathfrak{W}(\boldsymbol{\phi}_1) = 2$.

Corollary (Fiedler (1975))

The eigenfunction $\boldsymbol{\phi}_k$ affording λ_k has at most k positive weak nodal domains for $k \geq 1$. Consequently, $\mathfrak{W}(\boldsymbol{\phi}_k) \leq 2k$.

Discrete Nodal Domains ...

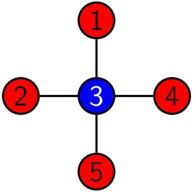
Corollary (Fiedler (1975))

If G is connected, then $\mathfrak{W}(\boldsymbol{\phi}_1) = 2$.

Corollary (Fiedler (1975))

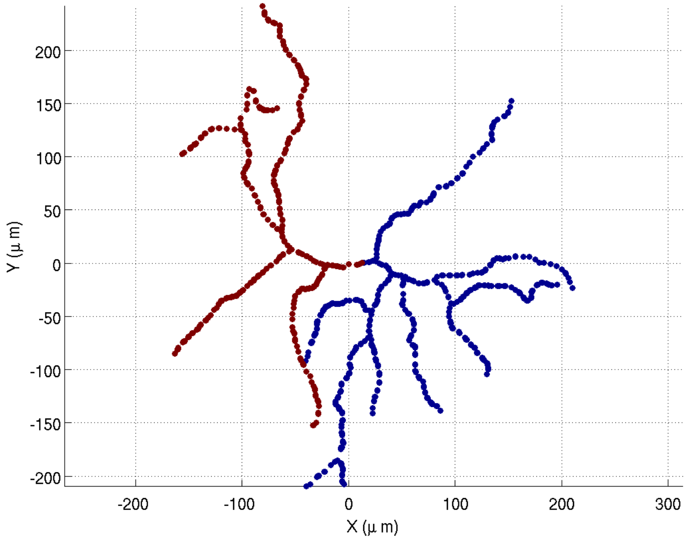
The eigenfunction $\boldsymbol{\phi}_k$ affording λ_k has at most k positive weak nodal domains for $k \geq 1$. Consequently, $\mathfrak{W}(\boldsymbol{\phi}_k) \leq 2k$.

In the previous example of $K_{1,4}$, we have $\lambda_{\max} = \lambda_4 = 5$, and $\boldsymbol{\phi}_4 \propto [1, 1, -4, 1, 1]^{\mathsf{T}}$. Hence, $\mathfrak{W}(\boldsymbol{\phi}_4) = 5 \leq 2 \cdot 4 = 8$, satisfying the corollary.

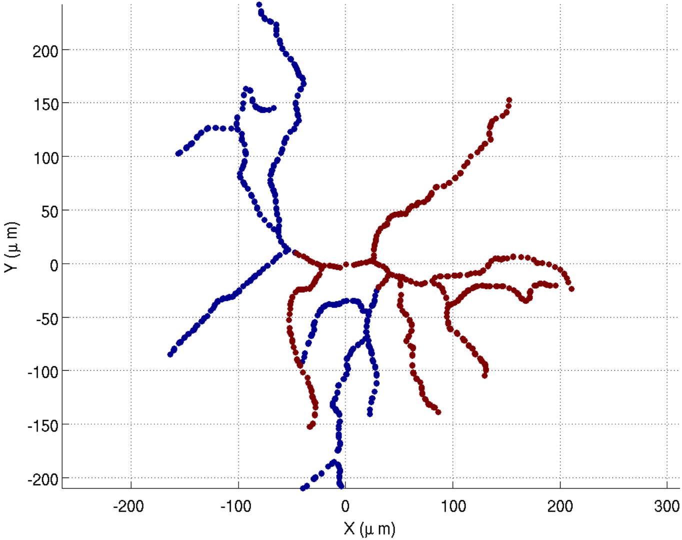


$K_{1,4}$

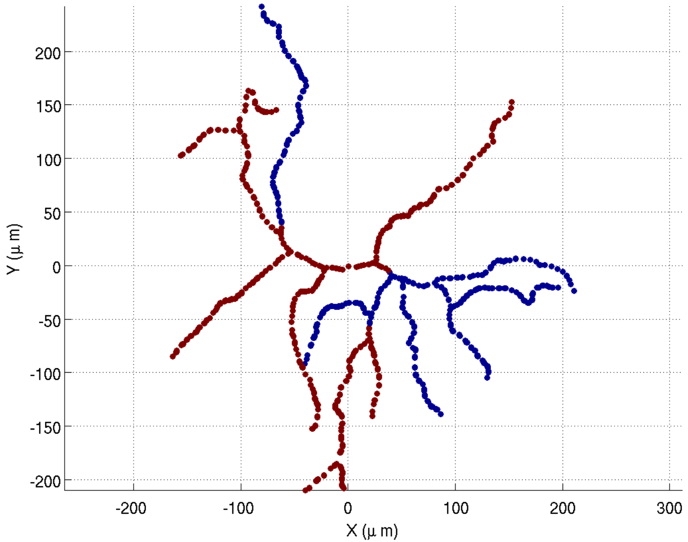
Discrete Nodal Domains of a Dendritic Tree: $\mathrm{sign}(\boldsymbol{\phi}_1)$



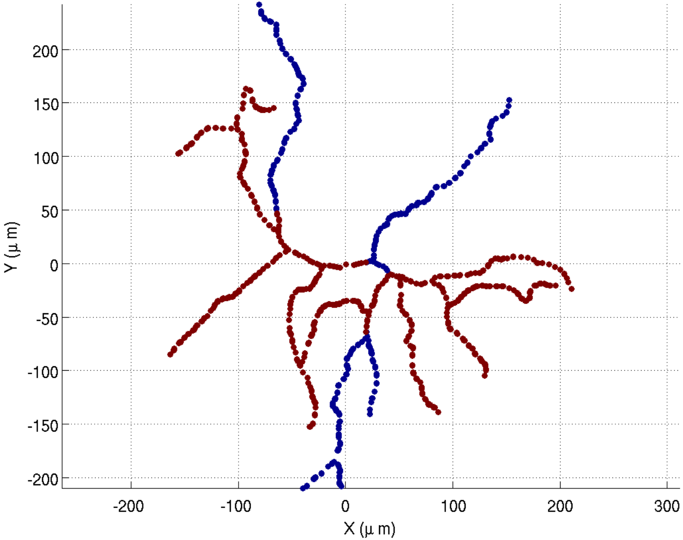
Discrete Nodal Domains of a Dendritic Tree: $\mathrm{sign}(\boldsymbol{\phi}_2)$



Discrete Nodal Domains of a Dendritic Tree: $\mathrm{sign}(\boldsymbol{\phi}_3)$



Discrete Nodal Domains of a Dendritic Tree: $\mathrm{sign}(\boldsymbol{\phi}_4)$



Outline

Introductory Remarks

- This part of my lecture is based on the following excellent tutorial paper:
  - U. von Luxburg: "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395-416, 2007.
- Spectral clustering has been successfully used in many applications, e.g., image and video segmentation, computer graphics, etc.; see e.g.,
  - J. Shi & J. Malik: "Normalized cuts and image segmentation", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 888–905, 2000.
  - S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, & J. C. Hart: "Spectral surface quadrangulation," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 1057-1066, 2006.

See also the references cited in von Luxburg's tutorial.

GL Eigenfunctions for L_{rw} and L_{sym}

Recall that we have three different versions of graph Laplacians:

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

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

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

Proposition (Properties of L_{rw} and L_{sym})

(a) $(\lambda, \boldsymbol{\phi})$ *is an eigenpair of* L_{rw} *iff* $(\lambda, D^{1/2}\boldsymbol{\phi})$ *is an eigenpair of* L_{sym}. *In particular,* $(0, \mathbf{1}_n)$ *for* $L_{\mathrm{rw}} \iff (0, D^{1/2}\mathbf{1}_n)$ *of* L_{sym}.

(b) $(\lambda, \boldsymbol{\phi})$ *is an eigenpair of* L_{rw} *iff* $(\lambda, \boldsymbol{\phi})$ *solves the generalized eigenproblem:* $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.

(c) *Both* L_{rw} *and* L_{sym} *are positive semi-definite and* n *nonnegative real-valued eigenvalues.*

GL Eigenfunctions for L_{rw} and L_{sym}

Recall that we have three different versions of graph Laplacians:

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

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

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

Proposition (Properties of L_{rw} and L_{sym})

(a) *$(\lambda, \boldsymbol{\phi})$ is an eigenpair of L_{rw} iff $(\lambda, D^{1/2}\boldsymbol{\phi})$ is an eigenpair of L_{sym}. In particular, $(0, \mathbf{1}_n)$ for $L_{\mathrm{rw}} \Longleftrightarrow (0, D^{1/2}\mathbf{1}_n)$ of L_{sym}.*

(b) *$(\lambda, \boldsymbol{\phi})$ is an eigenpair of L_{rw} iff $(\lambda, \boldsymbol{\phi})$ solves the generalized eigenproblem: $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.*

(c) *Both L_{rw} and L_{sym} are positive semi-definite and n nonnegative real-valued eigenvalues.*

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\Phi := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each row of Φ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth row vector of Φ.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\Phi := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each row of Φ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth row vector of Φ.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\Phi := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each row of Φ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth row vector of Φ.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\boldsymbol{\Phi} := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each row of $\boldsymbol{\Phi}$ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth row vector of $\boldsymbol{\Phi}$.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\boldsymbol{\Phi} := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each row of $\boldsymbol{\Phi}$ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth row vector of $\boldsymbol{\Phi}$.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\boldsymbol{\Phi} := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each row of $\boldsymbol{\Phi}$ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth row vector of $\boldsymbol{\Phi}$.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

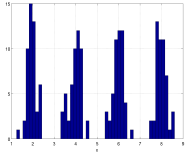
1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\boldsymbol{\Phi} := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each row of $\boldsymbol{\Phi}$ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth row vector of $\boldsymbol{\Phi}$.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Simple Examples for Spectral Clustering

- The following example was taken from Von Luxburg's tutorial paper with some modification.
- The dataset consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j$, $j = 1,2,3,4$, and $\sigma = 0.25$.
- These 200 points in $\mathbb{R}$ are the vertices in V.
- A complete graph K_{200} was generated with the edge weight by $a_{ij} = \exp(-|x_i - x_j|^2/2\epsilon^2)$ where $\epsilon = 1$ was used.
- Applied the spectral clustering algorithms.
- Note that we will discuss more about how to construct a graph from given datasets in the future lectures. The above strategy is used for simplicity.

Simple Examples for Spectral Clustering

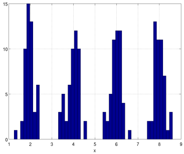
- The following example was taken from Von Luxburg's tutorial paper with some modification.
- The dataset consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j,\ j = 1,2,3,4$, and $\sigma = 0.25$.



- These 200 points in $\mathbb{R}$ are the vertices in V.
- A complete graph K_{200} was generated with the edge weight by $a_{ij} = \exp(-|x_i - x_j|^2/2\epsilon^2)$ where $\epsilon = 1$ was used.
- Applied the spectral clustering algorithms.
- Note that we will discuss more about how to construct a graph from given datasets in the future lectures. The above strategy is used for simplicity.

Simple Examples for Spectral Clustering

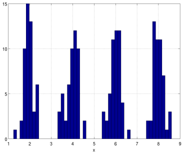
- The following example was taken from Von Luxburg's tutorial paper with some modification.
- The dataset consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j,\ j = 1,2,3,4$, and $\sigma = 0.25$.



- These 200 points in $\mathbb{R}$ are the vertices in V.
- A complete graph K_{200} was generated with the edge weight by $a_{ij} = \exp(-|x_i - x_j|^2/2\epsilon^2)$ where $\epsilon = 1$ was used.
- Applied the spectral clustering algorithms.
- Note that we will discuss more about how to construct a graph from given datasets in the future lectures. The above strategy is used for simplicity.

Simple Examples for Spectral Clustering

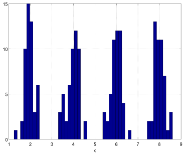
- The following example was taken from Von Luxburg's tutorial paper with some modification.
- The dataset consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j$, $j = 1, 2, 3, 4$, and $\sigma = 0.25$.



- These 200 points in $\mathbb{R}$ are the vertices in V.
- A complete graph K_{200} was generated with the edge weight by $a_{ij} = \exp(-|x_i - x_j|^2 / 2\epsilon^2)$ where $\epsilon = 1$ was used.
- Applied the spectral clustering algorithms.
- Note that we will discuss more about how to construct a graph from given datasets in the future lectures. The above strategy is used for simplicity.

Simple Examples for Spectral Clustering

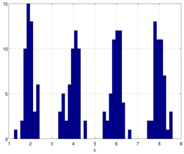
- The following example was taken from Von Luxburg's tutorial paper with some modification.
- The dataset consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j$, $j = 1, 2, 3, 4$, and $\sigma = 0.25$.



- These 200 points in $\mathbb{R}$ are the vertices in V.
- A complete graph K_{200} was generated with the edge weight by $a_{ij} = \exp(-|x_i - x_j|^2/2\epsilon^2)$ where $\epsilon = 1$ was used.
- Applied the spectral clustering algorithms.
- Note that we will discuss more about how to construct a graph from given datasets in the future lectures. The above strategy is used for simplicity.

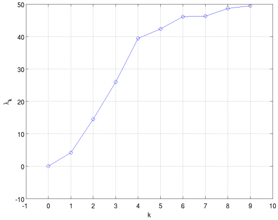
Simple Examples for Spectral Clustering

- The following example was taken from Von Luxburg's tutorial paper with some modification.
- The dataset consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j,\ j = 1, 2, 3, 4$, and $\sigma = 0.25$.

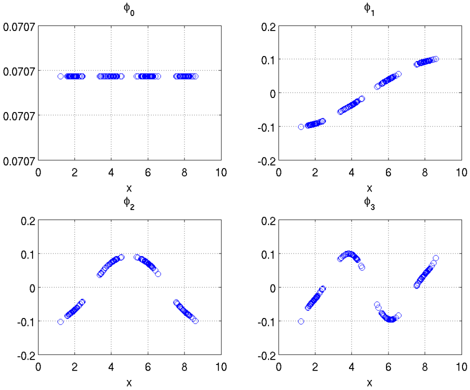


- These 200 points in $\mathbb{R}$ are the vertices in V.
- A complete graph K_{200} was generated with the edge weight by $a_{ij} = \exp(-|x_i - x_j|^2 / 2\epsilon^2)$ where $\epsilon = 1$ was used.
- Applied the spectral clustering algorithms.
- Note that we will discuss more about how to construct a graph from given datasets in the future lectures. The above strategy is used for simplicity.

Using L

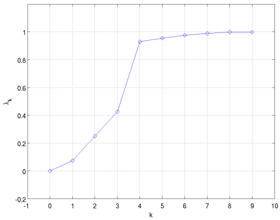


(a) λ_k

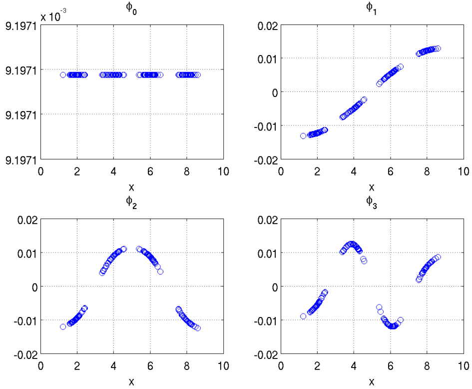


(b) $\boldsymbol{\phi}_k$

Using L_{rw}

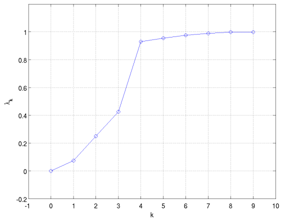


(a) λ_k

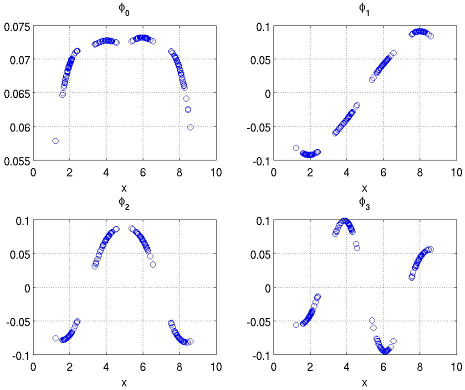


(b) $\boldsymbol{\phi}_k$

Using L_{sym}



(a) λ_k



(b) $\boldsymbol{\phi}_k$

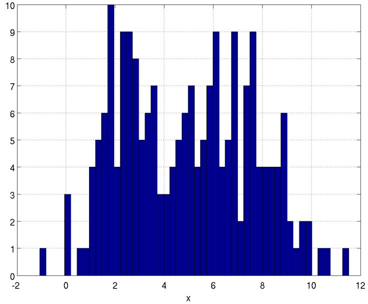
Simple Examples for Spectral Clustering ...

- Now, let's consider a less clear cut case. This time, the dataset still consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j$, $j = 1, 2, 3, 4$. But now I set the larger standard deviation, i.e., $\sigma = 1$ instead of $\sigma = 0.25$.

- Then let's repeat the same experiments and see how the situation changes.

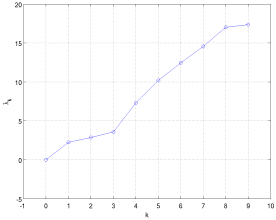
Simple Examples for Spectral Clustering . . .

- Now, let's consider a less clear cut case. This time, the dataset still consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j$, $j = 1, 2, 3, 4$. But now I set the larger standard deviation, i.e., $\sigma = 1$ instead of $\sigma = 0.25$.

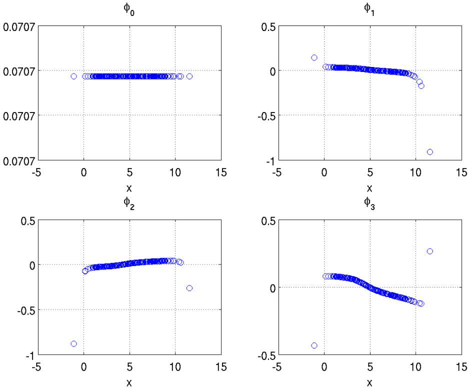


- Then let's repeat the same experiments and see how the situation changes.

Using L

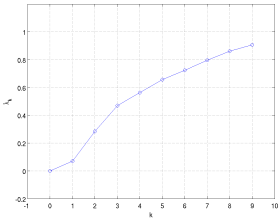


(a) λ_k

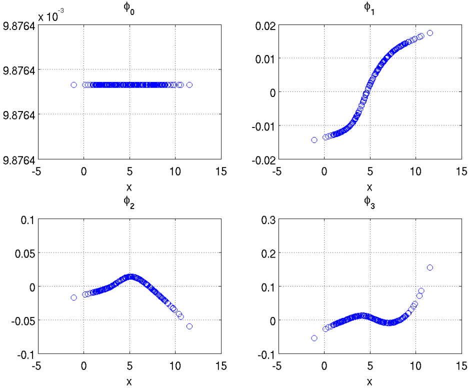


(b) $\boldsymbol{\phi}_k$

Using L_{rw}

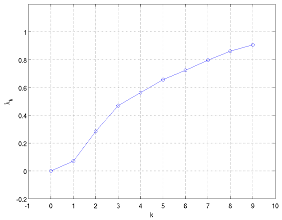


(a) λ_k

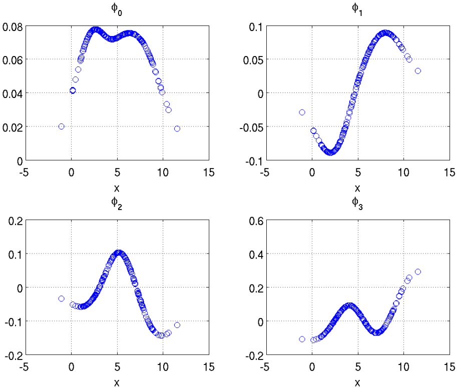


(b) $\boldsymbol{\phi}_k$

Using L_{sym}

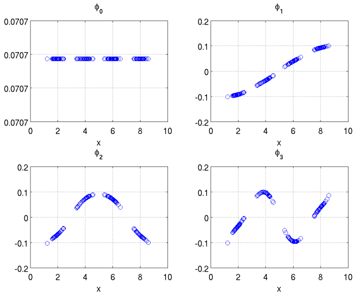


(a) λ_k

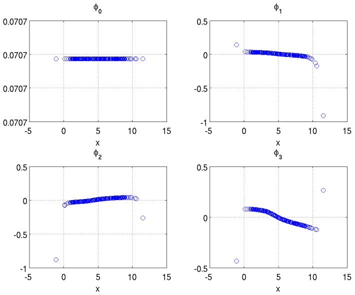


(b) $\boldsymbol{\phi}_k$

Using L

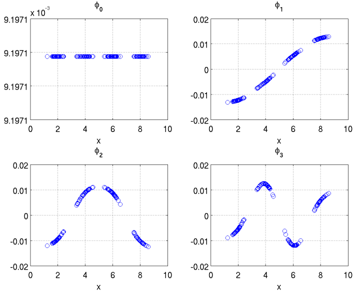


(a) $\sigma = 0.25$

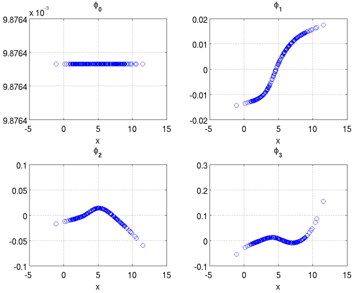


(b) $\sigma = 1$

Using L_{rw}

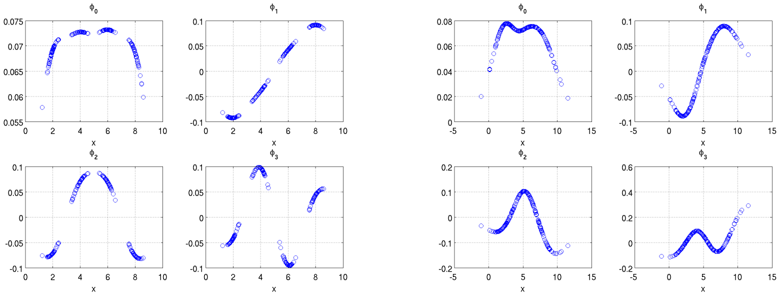


(a) $\sigma = 0.25$



(b) $\sigma = 1$

Using L_{sym}



(a) $\sigma = 0.25$ (b) $\sigma = 1$

Observations

- For the clear cut case, L, L_{rw}, and L_{sym} all performed similarly.
- Yet, the eigenvalue distributions of L_{rw} and L_{sym} revealed the number of existing clusters more clearly than that of L.
- For the case with severer overlaps, L_{rw} and L_{sym} outperformed L.
- von Luxburg recommends the use of L_{rw} because its ϕ_0 plays a role of the *cluster indicator vector*.

Observations

- For the clear cut case, L, L_{rw}, and L_{sym} all performed similarly.
- Yet, the eigenvalue distributions of L_{rw} and L_{sym} revealed the number of existing clusters more clearly than that of L.
- For the case with severer overlaps, L_{rw} and L_{sym} outperformed L.
- von Luxburg recommends the use of L_{rw} because its $\boldsymbol{\phi}_0$ plays a role of the *cluster indicator vector*.

Observations

- For the clear cut case, L, L_{rw}, and L_{sym} all performed similarly.
- Yet, the eigenvalue distributions of L_{rw} and L_{sym} revealed the number of existing clusters more clearly than that of L.
- For the case with severer overlaps, L_{rw} and L_{sym} outperformed L.
- von Luxburg recommends the use of L_{rw} because its ϕ_0 plays a role of the *cluster indicator vector*.

Observations

- For the clear cut case, L, L_{rw}, and L_{sym} all performed similarly.
- Yet, the eigenvalue distributions of L_{rw} and L_{sym} revealed the number of existing clusters more clearly than that of L.
- For the case with severer overlaps, L_{rw} and L_{sym} outperformed L.
- von Luxburg recommends the use of L_{rw} because its $\boldsymbol{\phi}_0$ plays a role of the *cluster indicator vector*.

Outline

Summary

- Provide an orthonormal basis on a general shape domain or a graph and allow spectral analysis/synthesis of data on them
- 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
- ∃ 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
- Connect to lots of interesting mathematics and applications: *harmonic analysis, discrete mathematics, mathematical physics, PDEs, differential geometry, signal & image processing, statistics, ...*

Summary

- Provide an orthonormal basis on a general shape domain or a graph and allow spectral analysis/synthesis of data on them
- 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
- ∃ 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
- Connect to lots of interesting mathematics and applications: *harmonic analysis, discrete mathematics, mathematical physics, PDEs, differential geometry, signal & image processing, statistics, . . .*

Summary

- Provide an orthonormal basis on a general shape domain or a graph and allow spectral analysis/synthesis of data on them
- 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
- ∃ 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
- Connect to lots of interesting mathematics and applications: *harmonic analysis, discrete mathematics, mathematical physics, PDEs, differential geometry, signal & image processing, statistics, ...*

Summary

- Provide an orthonormal basis on a general shape domain or a graph and allow spectral analysis/synthesis of data on them
- 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
- ∃ 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
- Connect to lots of interesting mathematics and applications: *harmonic analysis, discrete mathematics, mathematical physics, PDEs, differential geometry, signal & image processing, statistics, ...*

Summary

- Provide an orthonormal basis on a general shape domain or a graph and allow spectral analysis/synthesis of data on them
- 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
- ∃ 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
- Connect to lots of interesting mathematics and applications: *harmonic analysis, discrete mathematics, mathematical physics, PDEs, differential geometry, signal & image processing, statistics, ...*

Summary

- Provide an orthonormal basis on a general shape domain or a graph and allow spectral analysis/synthesis of data on them
- 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
- ∃ 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
- Connect to lots of interesting mathematics and applications: *harmonic analysis, discrete mathematics, mathematical physics, PDEs, differential geometry, signal & image processing, statistics, ...*

Summary

- Provide an orthonormal basis on a general shape domain or a graph and allow spectral analysis/synthesis of data on them
- 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
- $\exists$ 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
- Connect to lots of interesting mathematics and applications: *harmonic analysis, discrete mathematics, mathematical physics, PDEs, differential geometry, signal & image processing, statistics, ...*

Summary

- Provide an orthonormal basis on a general shape domain or a graph and allow spectral analysis/synthesis of data on them
- 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
- $\exists$ 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
- 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.
- N. Saito & E. Woei: "On the phase transition phenomenon of graph Laplacian eigenfunctions on trees," *RIMS Kôkyûroku*, vol. 1743, pp. 77–90, 2011.
- Y. Nakatsukasa, N. Saito, & E. Woei: "Mysteries around graph Laplacian eigenvalue 4," *Linear Algebra & Its Applications*, vol. 438, no. 8, pp. 3231–3246, 2013.

Thank you very much for your attention!