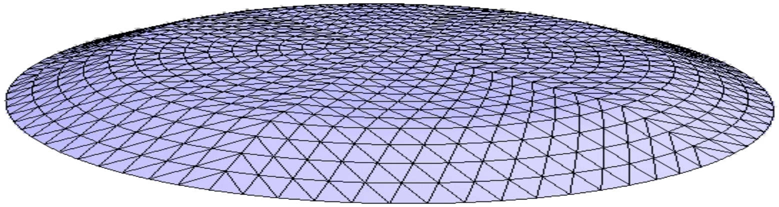


Laplace-Beltrami Eigenstuff Part 1 - Background

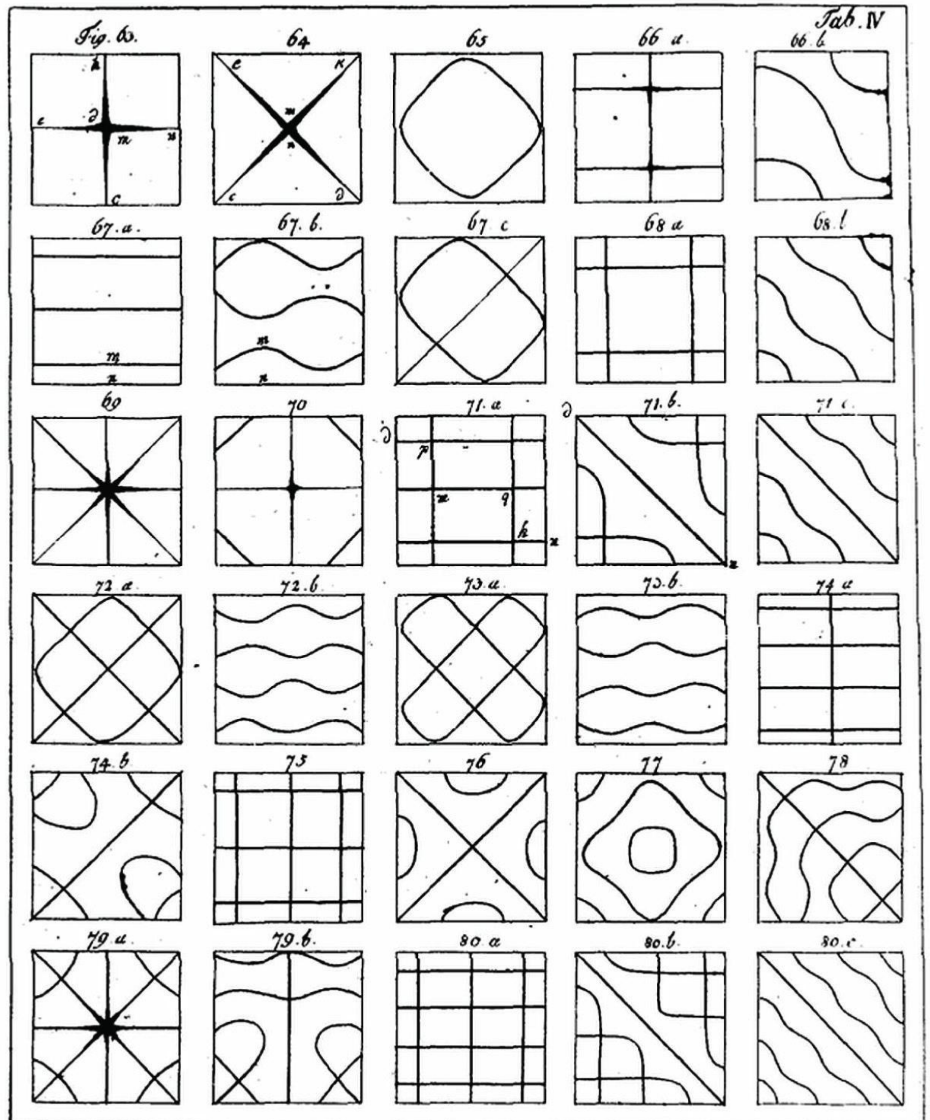
Martin Reuter – reuter@mit.edu
Mass. General Hospital, Harvard Medical, MIT

Sound and Shape

+

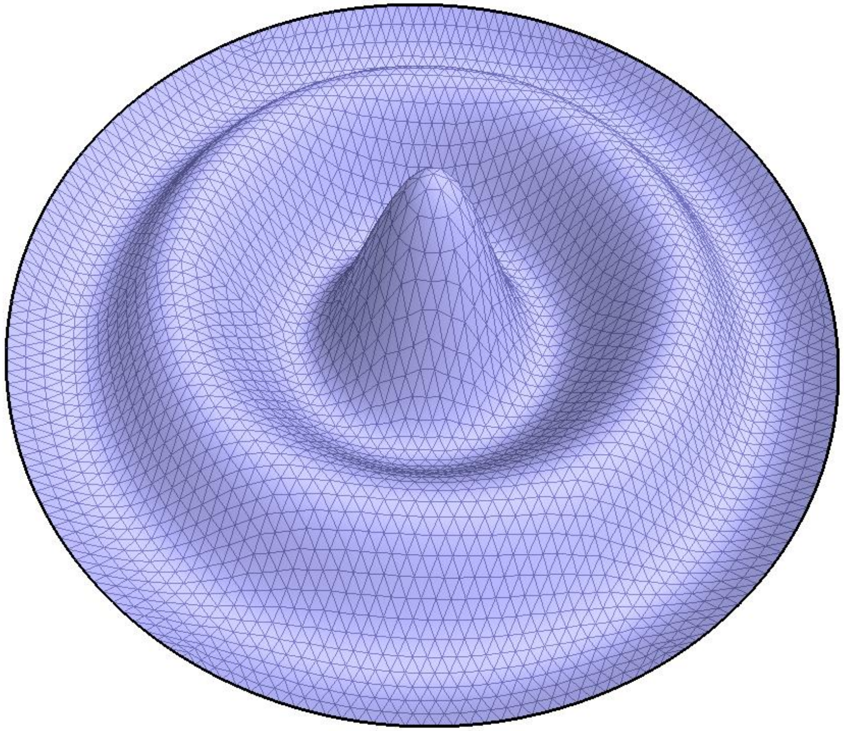
Chladni's strange patterns

- vibration of plates
- used a violin bow
- discovered sound patterns by spreading sand on the plates
- 1809 invited by Napoleon
- who held out price for mathematical explanation
- "Entdeckungen über die Theorie des Klanges" (Discoveries concerning the theory of music), Chladni, 1787



Can one "hear" Shape?

- First asked by Bers, then paper by Kac 1966, idea dates back to Weyl 1911 (at least).
- The sound (eigen-frequencies) of a drum depend on its shape.
- This spectrum can be numerically computed if the shape is known.
- E.g., no other shape has the same spectrum as a disk.
- Can the shape be computed from the spectrum?



Contributions

Shape Analysis: Reuter, Wolter, Peinecke [SPM05], [JCAD06](most cited paper award 09) and Patent Appl.

- Introduced Laplace-Beltrami Op. for Non-Rigid Shape Analysis.
- Cubic FEM to obtain accurate solutions for surfaces and solids.
- Before a mesh Laplace (simplified linear FEM) has been used for parametrization, smoothing, mesh compression

Image Recognition: Peinecke et al [JCAD07] (Mass Density LBO)

Neuroscience Applicantions:

- Statistical morphometric studies of brain structures (eigenvalues), Niethammer, Reuter, Shenton, Bioux.. [MICCAI07], Reuter.. [CW08]
- Topological studies of eigenfunctions, Reuter.. [CAD09] (invited)

Segmentation: Reuter..(IMATI, Genova) [SMI09] [IJCV09]

Correspondence: Reuter [IJCV09]

Why is the Laplace Operator so interesting?

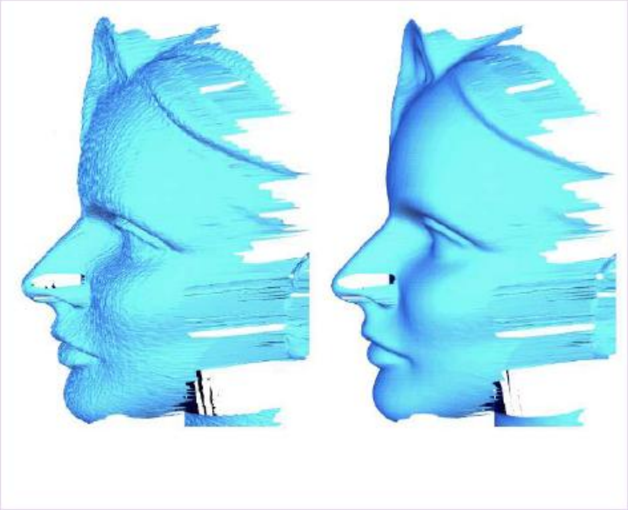
involved in many PDE's:

- Poisson equation: $\Delta f = h$
- Laplace equation: $\Delta f = 0$
- Heat equation: $\frac{\partial f}{\partial t} - k\Delta f = 0$
- Wave equation: $\frac{\partial^2 f}{\partial t^2} - k\Delta f = 0$
- Helmholtz equation: $\Delta f = -\lambda f$

=> Many properties carry over to potential applications from Physics …

Applications

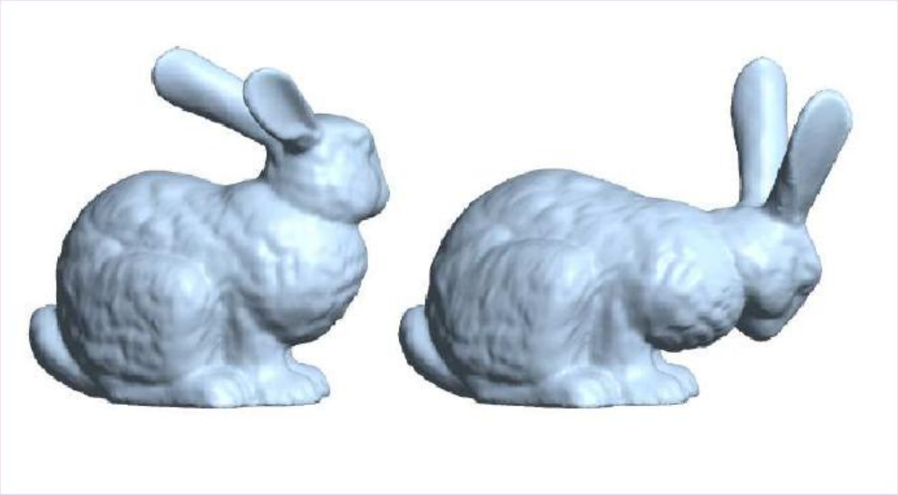
Mesh Smoothing



[Desbrun et al 1999, etc]

Applications

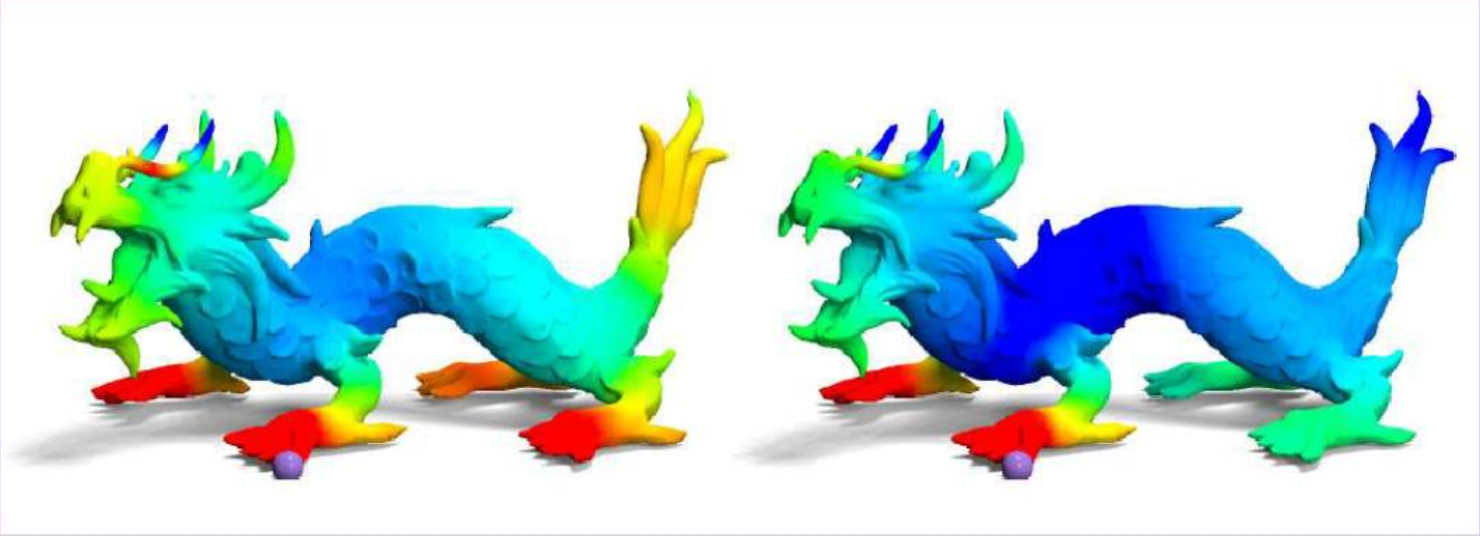
Mesh Editing



[Zhou et al 2005, Lipman et al 2005, etc]

+
Applications

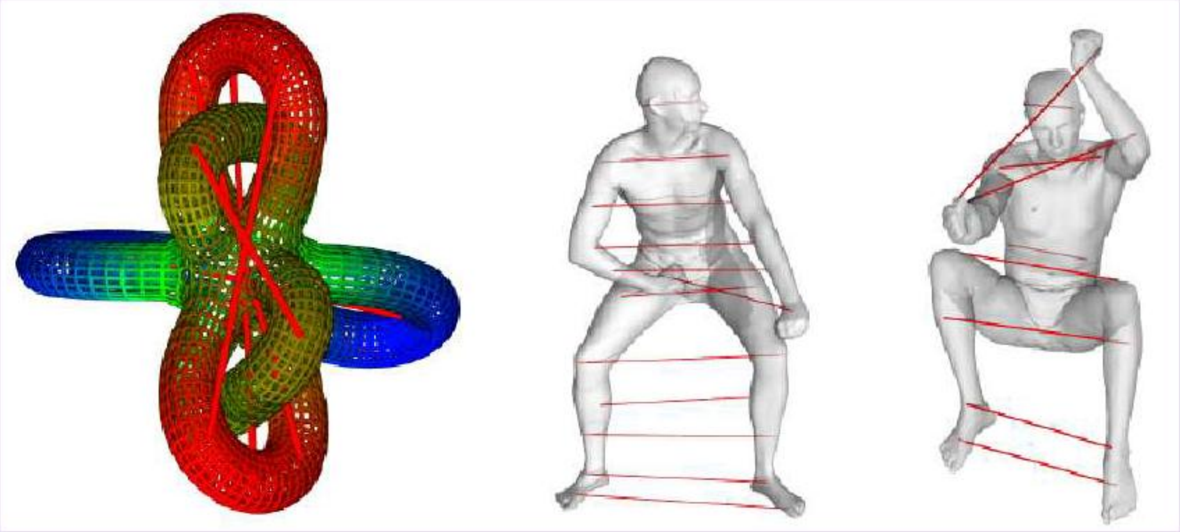
Heat Kernel Signature



[Sun, Ovsjanikov, and Guibas 2008, etc]

Applications

Shape Analysis



[Ovsjanikov, Sun and Guibas 2008, etc]

Laplace Spectrum

Definition

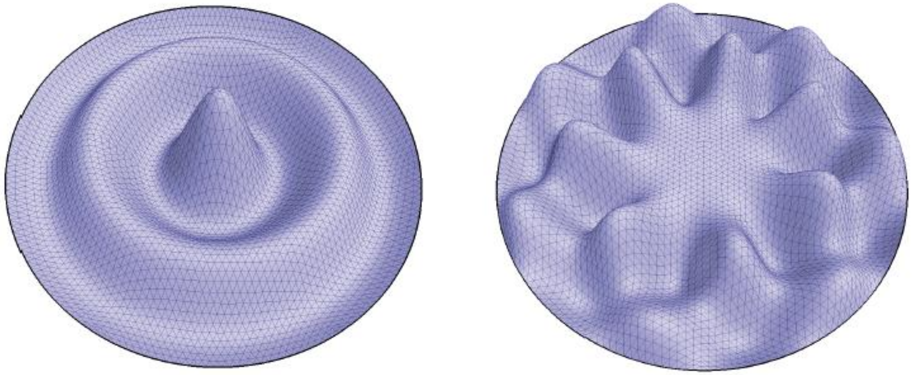
Helmholtz Equation (**Laplacian Eigenvalue Problem**):

$$\Delta f = -\lambda f, \qquad f : M \to \mathbb{R}$$

Solution: Eigenfunctions f_i with corresponding family of eigenvalues (**Spectrum**):

$$0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \uparrow +\infty$$

Here Laplace-Beltrami Operator: $\Delta f := div(grad\ f)$



Boundary Conditions

Laplace-Beltrami Spectrum for Manifolds with Boundary:

Dirichlet Boundary Condition

Function is fixed $f \equiv 0$ on the boundary of M

Neumann Boundary Condition

Derivative in normal direction is fixed $\frac{\partial f}{\partial n} \equiv 0$ on the boundary of M

1-Dimensional Helmholtz

$$\Delta f = -\lambda f \quad \Omega = \{x : 0 \le x \le a\}$$

$$\Leftrightarrow f'' = -\lambda f \Leftrightarrow f'' + \lambda f = 0$$

Neumann Boundary Condition :

$$f'(0) = 0 \quad and \quad f'(a) = 0$$

$$\Rightarrow f(x) = c \cdot \cos\left(\frac{n\pi}{a} x\right) \quad n = 0, 1, 2, \ldots$$

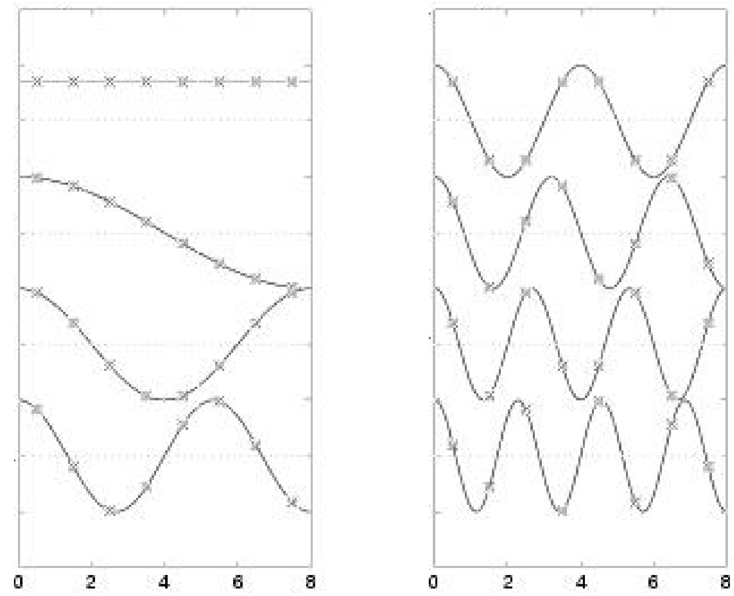
$$\Rightarrow \lambda = \left(\frac{n\pi}{a}\right)^2$$

Dirichlet Boundary Condition :

$$f(0) = 0 \quad and \quad f(a) = 0$$

$$\Rightarrow f(x) = c \sin(n\pi x / a) \quad n = 1, 2, 3, \ldots$$

$$\Rightarrow \lambda = (n\pi / a)^2$$



2D Rectangle

- Side lengths a and b (Neumann Boundary Condition)
- Separation of variables leads to Eigenfunctions:

$$f(x,y) = c\cos\left(\frac{m\pi}{a}x\right)\cos\left(\frac{n\pi}{b}x\right) \qquad m,n = 0,1,2,3,\cdots$$

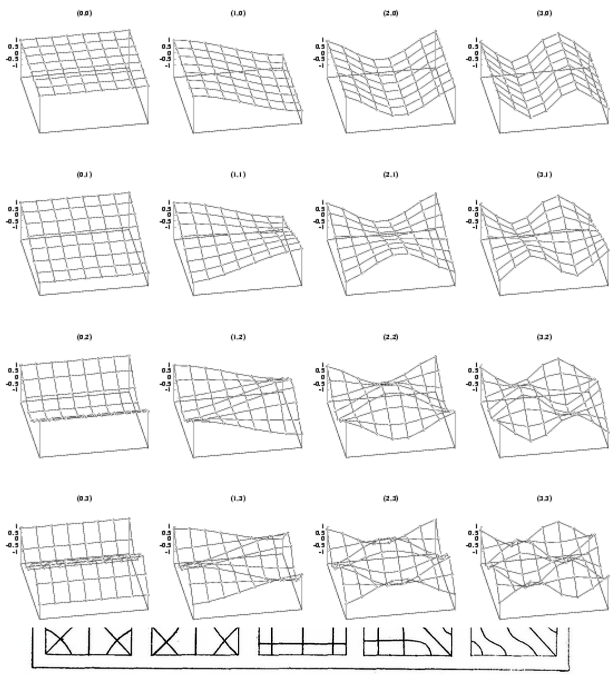
- Eigenvalues:

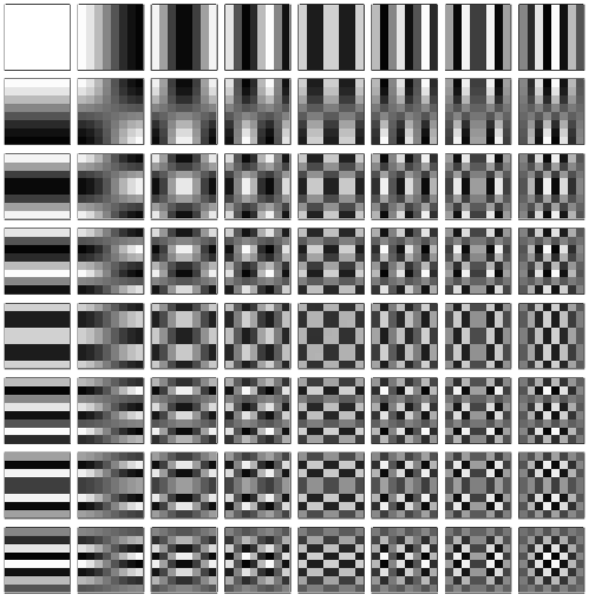
$$\lambda = \pi^2\left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)$$

- Other cases where spectrum is known analytically:
  - Circle (Bessel functions)
  - Cylinder, flat torus (basically rectangle with special bndr. cond.)
  - Sphere (Spherical Harmonics)

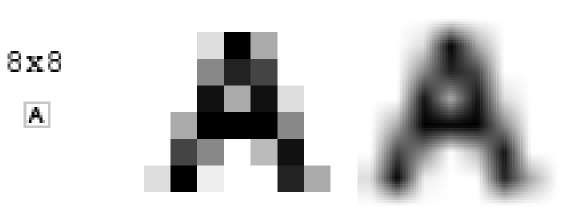
Discrete Cosine Transform

- Similar to Fourier Transform, but only cosines at different frequencies to combine a signal.
- Compression
  - MP3
  - JPEG

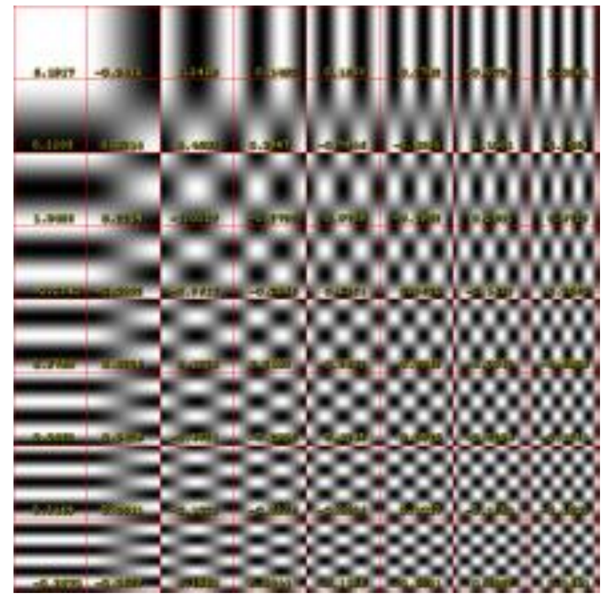


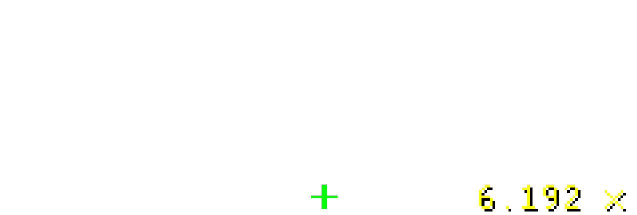


DCT at Work



$$\begin{bmatrix} 6.1917 & -0.3411 & 1.2418 & 0.1492 & 0.1583 & 0.2742 & -0.0724 & 0.0561 \\ 0.2205 & 0.0214 & 0.4503 & 0.3947 & -0.7846 & -0.4391 & 0.1001 & -0.2554 \\ 1.0423 & 0.2214 & -1.0017 & -0.2720 & 0.0789 & -0.1952 & 0.2801 & 0.4713 \\ -0.2340 & -0.0392 & -0.2617 & -0.2866 & 0.6351 & 0.3501 & -0.1433 & 0.3550 \\ 0.2750 & 0.0226 & 0.1229 & 0.2183 & -0.2583 & -0.0742 & -0.2042 & -0.5906 \\ 0.0653 & 0.0428 & -0.4721 & -0.2905 & 0.4745 & 0.2875 & -0.0284 & -0.1311 \\ 0.3169 & 0.0541 & -0.1033 & -0.0225 & -0.0056 & 0.1017 & -0.1650 & -0.1500 \\ -0.2970 & -0.0627 & 0.1960 & 0.0644 & -0.1136 & -0.1031 & 0.1887 & 0.1444 \end{bmatrix}$$





Source: Wikipedia

Heat Kernel

- Heat Equation:
  - Temperature distribution after time t:

$$\frac{\partial}{\partial t} f + \Delta f = 0$$

- Solution (with initial condition f_0):

$$f(\vec{x}, t) = \int_M K(\vec{x}, \vec{y}, t) f_0(\vec{y}) d\vec{y}$$

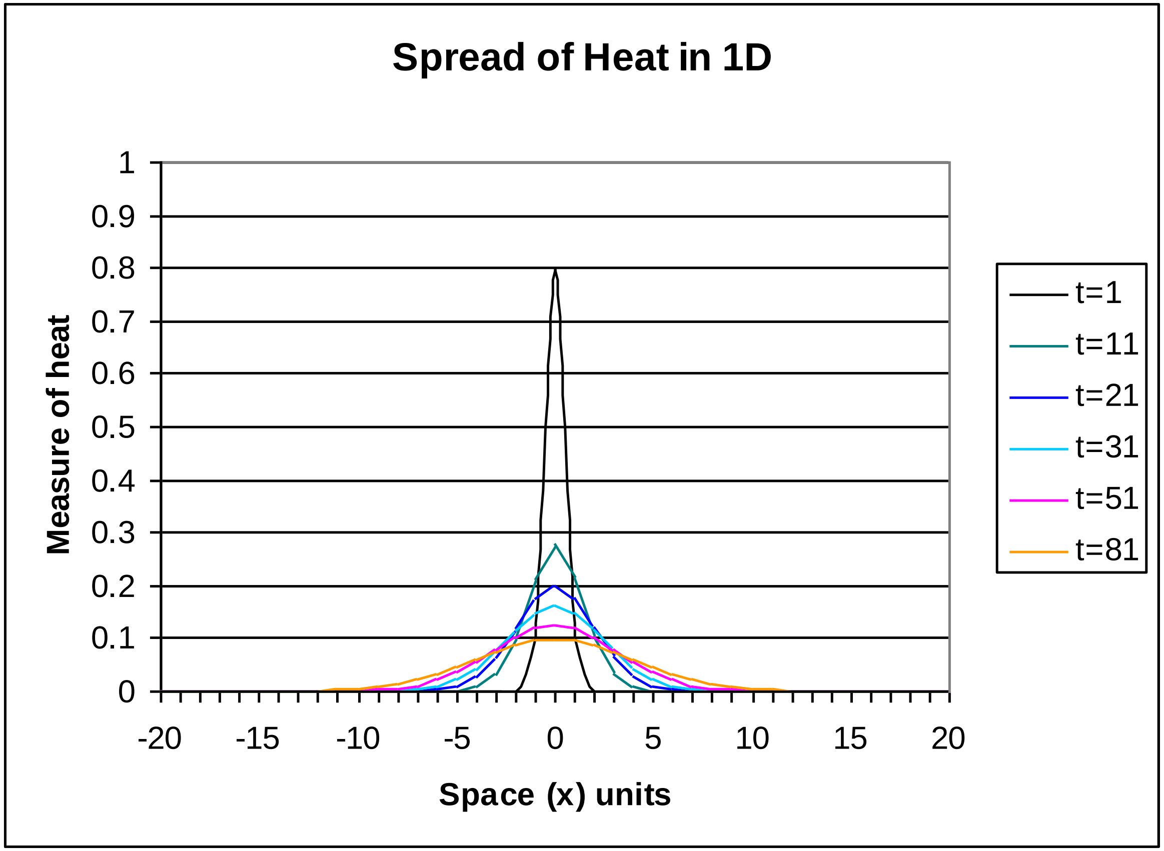
- Initial condition unit heat source at point p:

$$f(\vec{x}, t) = K(\vec{x}, \vec{p}, t)$$

- Once we know Eigenfunctions and –values:

$$K(\vec{x}, \vec{y}, t) = \sum_{n=0}^{\infty} \exp^{-\lambda_n t} f_n(\vec{x}) f_n(\vec{y})$$

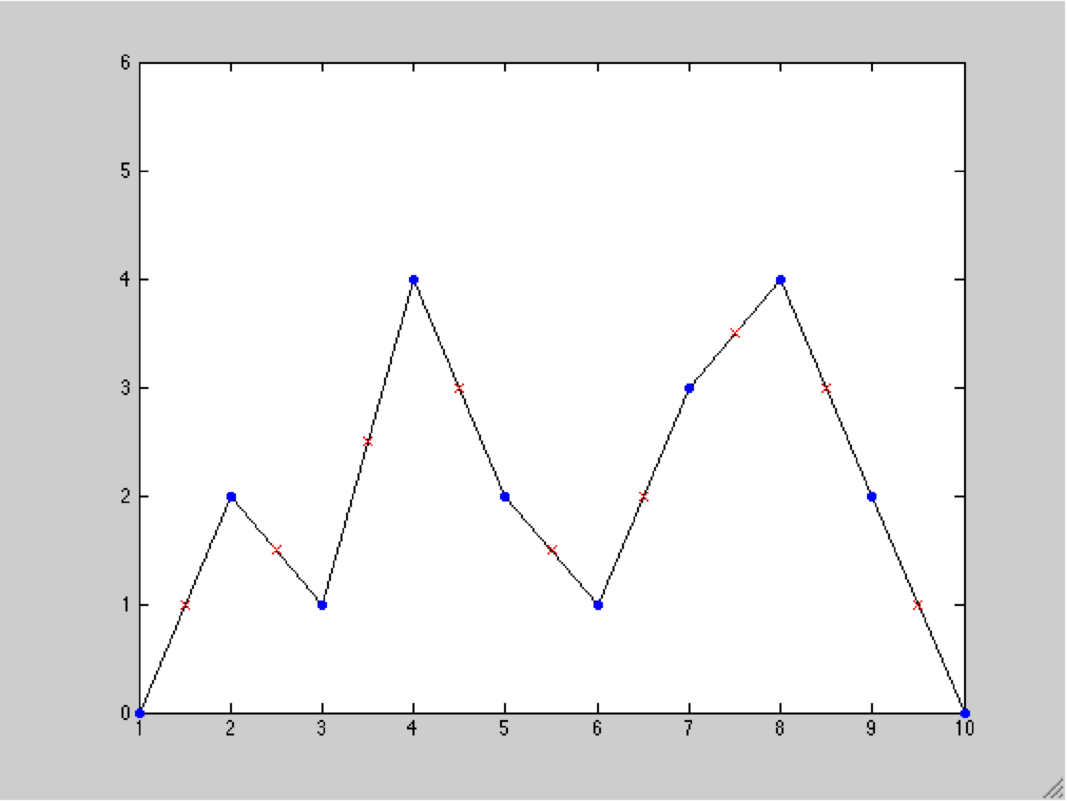
1D Heat Kernel



Discrete 1D Laplacian

The Laplace operator is simply the second derivative:

$$\Delta f := f'' \quad in \quad °$$



$$\Delta y_i = \left(\frac{y_{i+1} - y_i}{h} - \frac{y_i - y_{i-1}}{h} \right) \frac{1}{h}$$

$$= \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} = \sum_{j \in \aleph(i)} \frac{y_i - y_j}{h^2}$$

$$h = 1 \Rightarrow \begin{pmatrix} 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} y_{i-1} \\ y_i \\ y_{i+1} \end{pmatrix} \quad (1D\ filter)$$

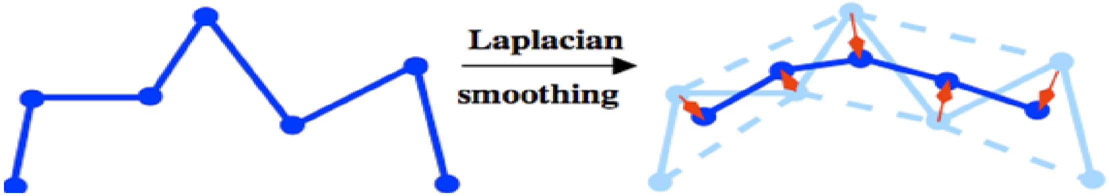
$$Full: \begin{pmatrix} \ldots -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2\ldots \end{pmatrix} y = Ly$$

$$W = adj(G) \quad and \quad V_{ii} = \#Neighbors$$

$$L = W - V$$

1D-Laplacian-Smoothing

Move vertex half way towards center of neighbors:



$$\hat{\mathbf{v}}_i = \frac{1}{2}\left[\frac{1}{2}(\mathbf{v}_{i-1} + \mathbf{v}_i)\right] + \frac{1}{2}\left[\frac{1}{2}(\mathbf{v}_i + \mathbf{v}_{i+1})\right] = \frac{1}{4}\mathbf{v}_{i-1} + \frac{1}{2}\mathbf{v}_i + \frac{1}{4}\mathbf{v}_{i+1}$$

$$\hat{v}_i = v_i + \frac{1}{4}\Delta v_i \quad where \quad \Delta v_i = \begin{pmatrix} \Delta x_i \\ \Delta y_i \end{pmatrix}$$

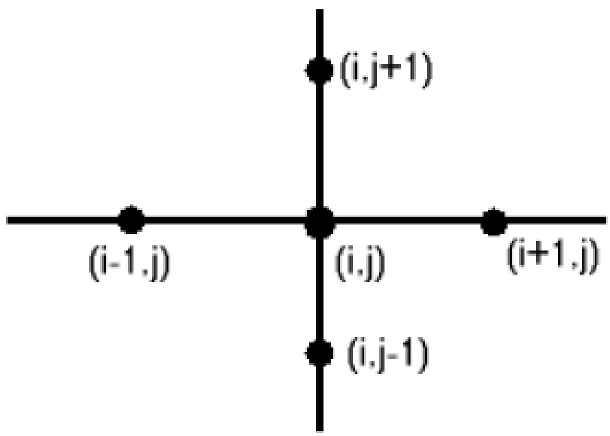
…but don't wash it too long, it will shrink!

2D Grid Laplacian

In 2D Laplace is sum of second partial derivatives:

$$\Delta f(x,y) := f_{xx} + f_{yy} \quad in \quad \circ^{2}$$

Discrete 2D-Filter is a 5 Stencil:



$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$again \quad L = W - V$$

Graph Laplacian

Extension to a Graph is simply (guess):

$$L = W - V \quad or \quad \Delta y_i = \sum_{j \in \aleph(i)} (y_j - y_i)$$

With (symmetric) edge weights (guess again):

$$\Delta y_i = \sum_{j \in \aleph(i)} w_{ij}(y_j - y_i) \quad or \quad L := W - V \quad with \quad W := (w_{ij}) \quad and \quad V := diag\left(\sum_{j \in \aleph(i)} w_{ij} \right)$$

And finally with node weights:

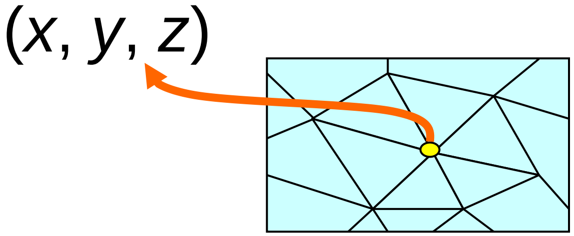
$$\Delta y_i = \frac{1}{d_i} \sum_{j \in \aleph(i)} w_{ij}(y_j - y_i) \quad or \quad L := D^{-1}(W - V) \quad with \quad D := diag(d_i)$$

Spectral Graph Theory

- Eigenvalues
  - Closely related to variety of global graph invariants
  - Global shape descriptors
- Eigenvectors
  - Useful extremal properties, e.g., heuristic for NP-hard problems, normalized cuts and sequencing
  - Spectral embeddings capture global information, e.g., clustering, manifold learning
- However, we'll not look at graphs here, but meshes …

Meshes

Meshes can be seen as graphs:



… with a geometry (here triangle mesh embedded into R3).

Therefore one possibility is to use same Laplace discretizations (with appropriate weights)

How to choose weights “correctly” ?

Operator Discretizations

- Graph Laplace:
  - Vertex weights =1 (masses) and edge weights = 1 (stiffness)
  - Other options (vertex degree, other symmetric weights)
  - Usually not geometry aware
- Mesh Laplace:
  - Different weights (based on geometry)
  - Possibly mesh dependent
  - What if we go to quad-meshes
  - Or tetrahedral meshes (3D)…
- FEM Laplace:
  - Different approach based on Surface or Volume elements
  - Generalizable and higher order approximations

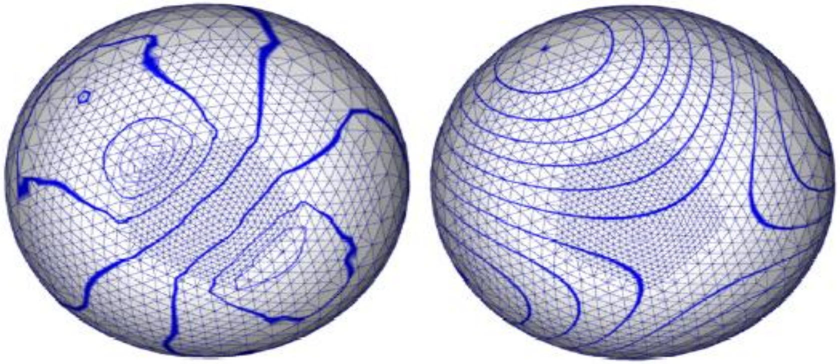


Figure courtesy of Bruno Levy

Mesh Laplacians (Examples)

- Pinkall and Polthier (93): $w_{ij} = \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2}$ $d_i = 1$

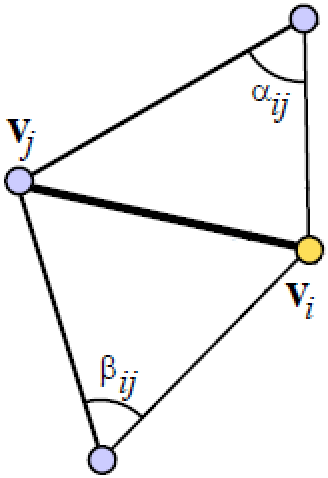
  where α and βdenote the two angles opposite edge(i,j). Still depends on mesh sampling (missing mass weights).

- Desbrun (99): $d_i = \frac{area_i}{3}$

  where the area is summed for all triangles at vertex i. Will turn out to be a simplified linear finite element operator.

- Meyer (02): $d_i = \frac{voronoi_i}{3}$

  use the the area obtained by joining the circumcenters of the triangles around vertex i (i.e., the Voronoi region).



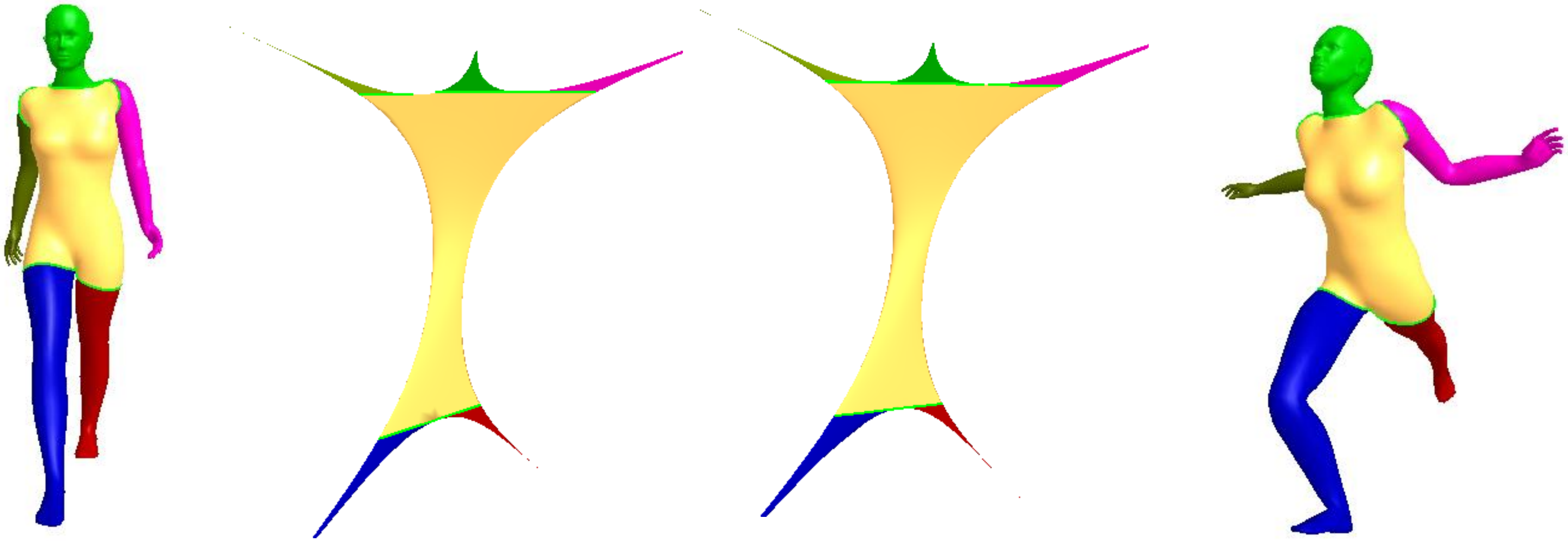
Finite Element Method

- Instead of graph/wireframe (vertices, edges), we look at elements that assemble our geometry without gaps:
  - triangles
  - tetrahedra
  - voxels….
- We define basis functions over this discretized geometry (linear, quadratic, cubic …)
- We get a powerful framework to solve differential equations (not just Laplace).
- Details later…

Spectral Embedding

- Maps each vertex to the value of all (or a few) Eigenfunctions at that location:

$$\Phi(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x}), \ldots)$$



Spectral Embedding

- Dirichlet Energy measures smoothness
- For Eigenfunctions it is:

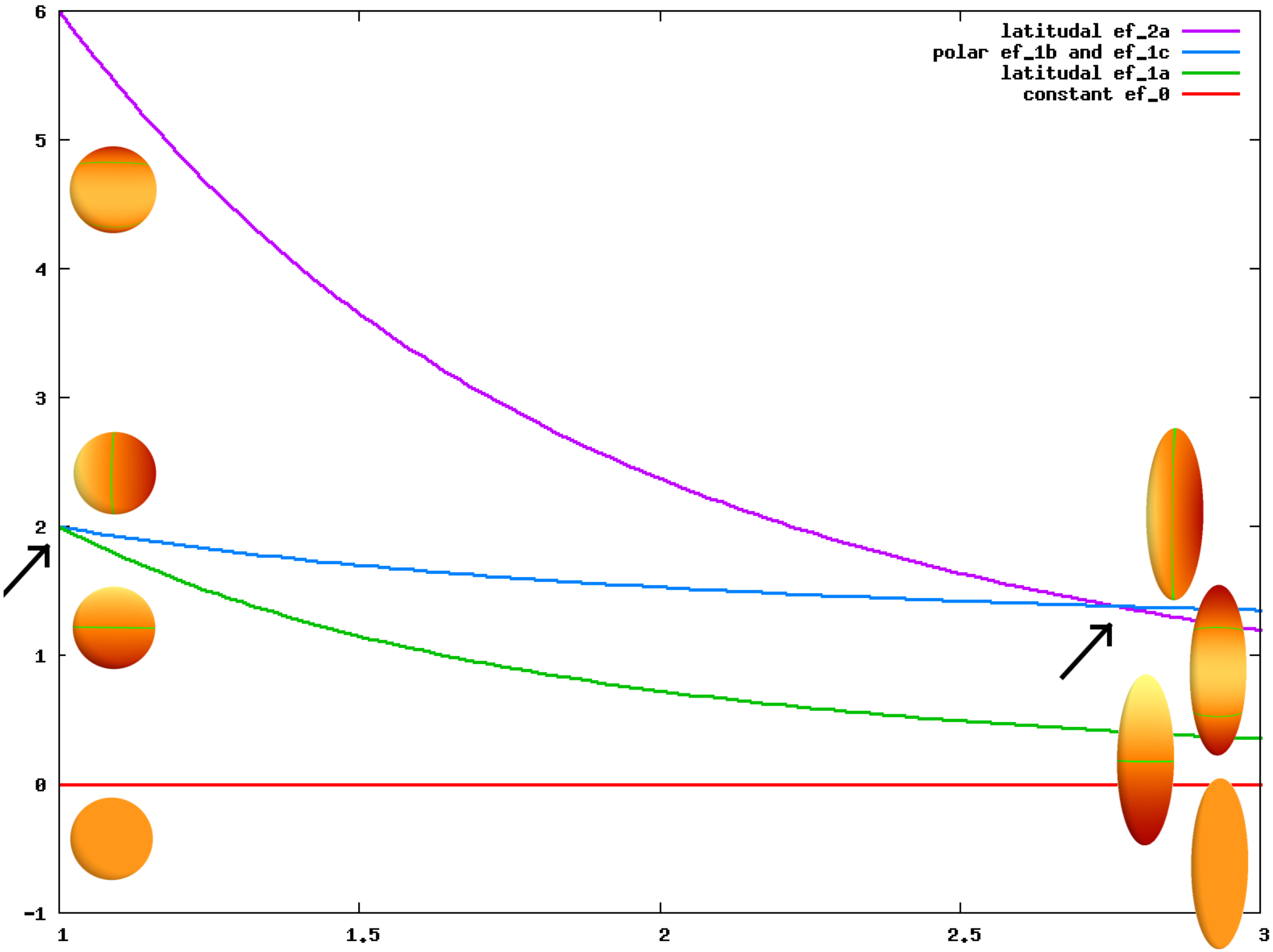
$$\begin{aligned} E[f_i] &= \int_M |\nabla f_i|^2 \, d\sigma = \int_M (\nabla f_i)^2 \, d\sigma \\ &= -\int_M f_i \Delta f_i \, d\sigma = \lambda_i \int_M f_i f_i \, d\sigma = \lambda_i \end{aligned}$$

- Optimal Embedding:
  - The first (non constant) Eigenfunction (also called Fiedler vector) yields the optimal (smoothest) embedding of the shape onto a line (orthogonal to constant function and complying with boundary condition)
  - Higher functions yield smoothest embedding orthogonal to previous ones

Difficulties

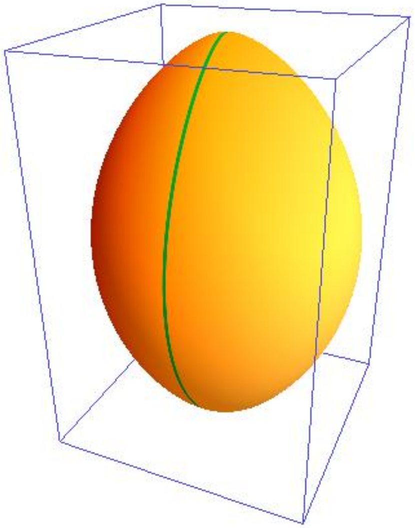
- Numerical inaccuracies
- Sign Flips
  - Scalar multiples are contained in same space
- Higher dimensional Eigenspaces
  - Arbitrary basis (luckily they are rare, but being close to one is already problematic)
  - Due to numerical errors Eigenvalues will be slightly different and we cannot really detect these situations
- Switching of Eigenfunctions
  - Occurs, because of numerical instabilities (two close Eigenvalues switch)
  - Or due to geometric (non-isometric) deformations

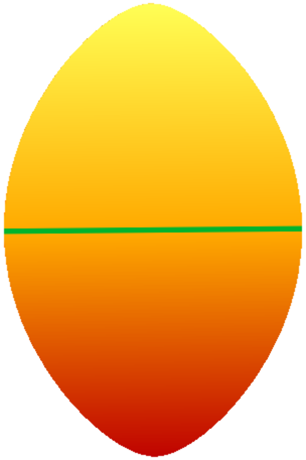
Switching of Eigenfunctions



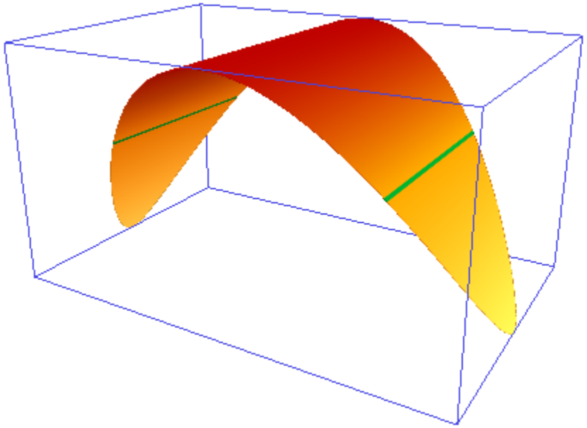
Switching of Eigenfunctions

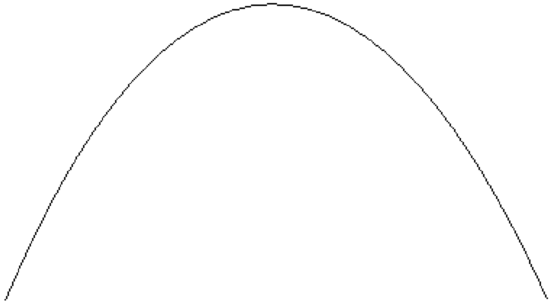
- Z=2.74





- Z=2.76





+

Applications: smoothing revisited

- Represent any function as linear combination of Eigenfunctions (similar to DCT):

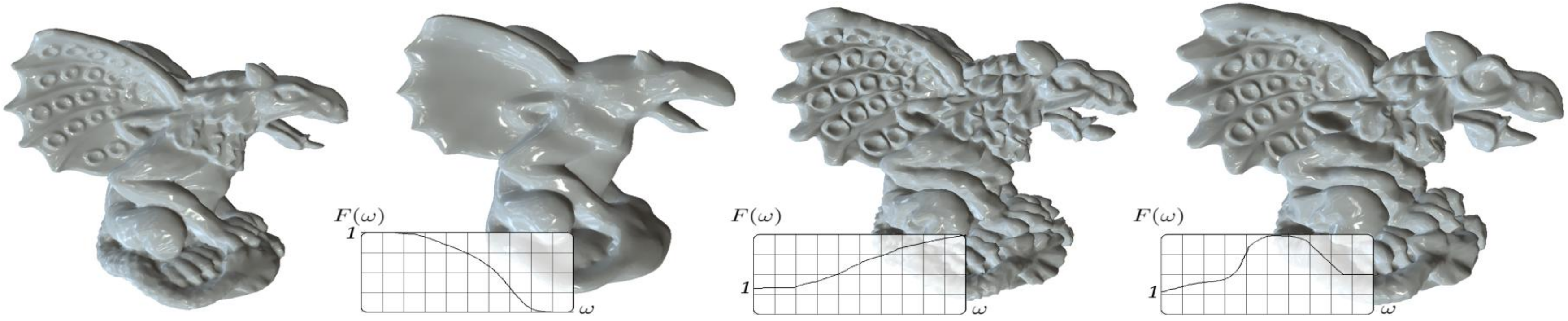
$$F(\vec{x}) = \sum_{i=1}^{N} c_i f_i$$

- Smooth iteratively (modify coefficients):

$$S^m F = (I + \frac{1}{4}L)^m F = \sum_{i=1}^{n} (I + \frac{1}{4}L)^m c_i f_i = \sum_{i=1}^{n} \underbrace{(1 - \frac{1}{4}\lambda_i)^m}_{\text{filter}} c_i f_i$$

Geometry Filtering

Take **f** =(x,y,z)



Courtesy of Bruno Levy

Color Filtering on Mesh

Take **f** =(r,g,b)

Courtesy of Bruno Levy