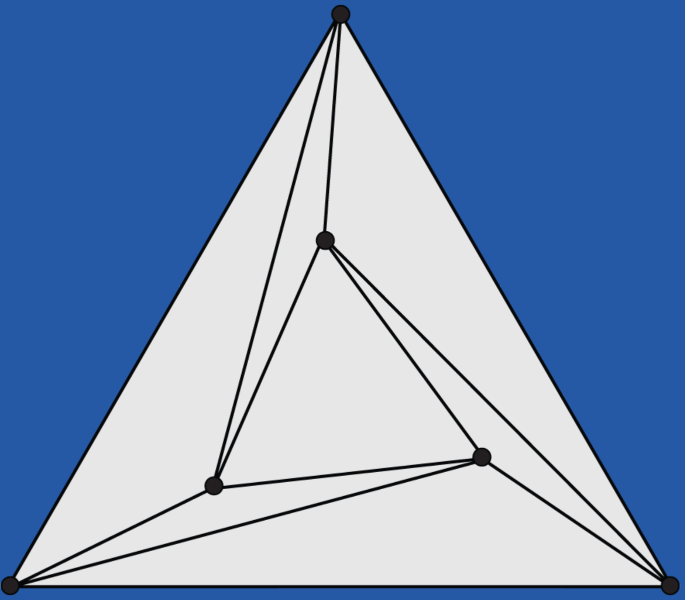
A Primer on Laplacians



Max Wardetzky

Institute for Numerical and Applied Mathematics

Georg-August Universität Göttingen, Germany

Warm-up: Euclidean case

Warm-up – The Euclidean case

Chladni's vibrating plates

$$\Delta u = \lambda u$$

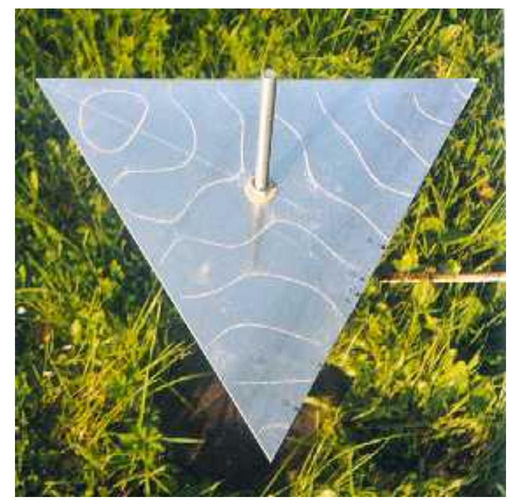


Plate vibrated by violin bow

Sand settles on nodal curves

Warm-up – The Euclidean case

$$\Delta u = -\left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \cdots + \frac{\partial^2 u}{\partial x_n^2}\right)$$

Basic properties:

$$(u, \Delta v) = \int_\Omega \nabla u \cdot \nabla v = (\Delta u, v) \qquad \text{(Sym)}$$

$$(u, \Delta u) = \int_\Omega \nabla u \cdot \nabla u \geq 0 \qquad \text{(Psd)}$$

$$\Delta u = \lambda u \Rightarrow \lambda \geq 0$$

Warm-up – The Euclidean case

$$\Delta u = -\left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \cdots + \frac{\partial^2 u}{\partial x_n^2}\right)$$

Basic properties:

$$(u, \Delta v) = \int_\Omega \nabla u \cdot \nabla v = (\Delta u, v) \qquad \text{(Sym)}$$

$$(u, \Delta u) = \int_\Omega \nabla u \cdot \nabla u \geq 0 \qquad \text{(Psd)}$$

$$u \text{ harmonic} \Rightarrow \text{no strict loc. max. in } \Omega \qquad \text{(Max)}$$

Riemannian case

Laplacians on Riemannian manifolds

$$\Delta u = -\mathrm{div}\,\nabla u \qquad \left(\int_\Omega \nabla u \cdot X = \int_\Omega u\,(-\mathrm{div} X)\right)$$

Using exterior differentiation:

$$\int_U d\alpha = \int_{\partial U} \alpha$$

k-form

(Stokes for k-forms)

(k+1)-form

Laplacians on Riemannian manifolds

$$\Delta u = -\mathrm{div}\,\nabla u \qquad \left(\int_\Omega \nabla u \cdot X = \int_\Omega u\,(-\mathrm{div} X)\right)$$

Using exterior differentiation:

$$(\alpha, \beta)_k := \int_M g(\alpha, \beta)\mathrm{vol}_g$$ (inner product on k-forms)

$$(d\alpha, \beta)_{k+1} = (\alpha, d^*\beta)_k$$ (adjoint to d-operator)

$$\Delta\alpha := dd^*\alpha + d^*d\alpha$$ (Laplacian on k-forms)

Laplacians on Riemannian manifolds

$$\Delta u = -\mathrm{div}\,\nabla u \qquad \left(\int_\Omega \nabla u \cdot X = \int_\Omega u\,(-\mathrm{div} X)\right)$$

Using exterior differentiation:

$$(\alpha, \beta)_k := \int_M g(\alpha, \beta)\mathrm{vol}_g$$ (inner product on k-forms)

$$(d\alpha, \beta)_{k+1} = (\alpha, d^*\beta)_k$$ (adjoint to d-operator)

$$\Delta u = d^* du$$ (Laplacian on functions)

Laplacians on Riemannian manifolds

$$\Delta u = d^* du$$

(Laplacian on functions)

Basic properties:

$$(u, \Delta v) = \int_M g(du, dv) = (\Delta u, v) \qquad \text{(Sym)}$$

$$(u, \Delta u) = \int_M g(du, du) \geq 0 \qquad \text{(Psd)}$$

$$u \text{ harmonic} \Rightarrow \text{no strict loc. max. in } \Omega \qquad \text{(Max)}$$

Why should we care?

Three reasons…

1. Topology

Hodge-Helmholtz decomposition of k-forms:

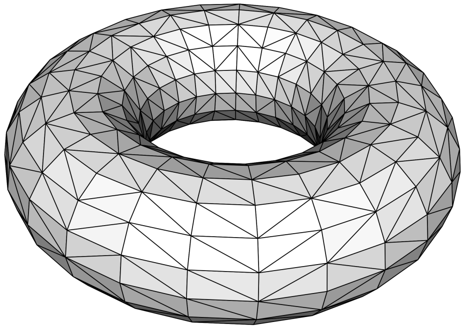
$$\alpha = d\mu + d^*\nu + h$$

(unique and L^2-orthogonal)

$$\boxed{\Delta h = 0}$$

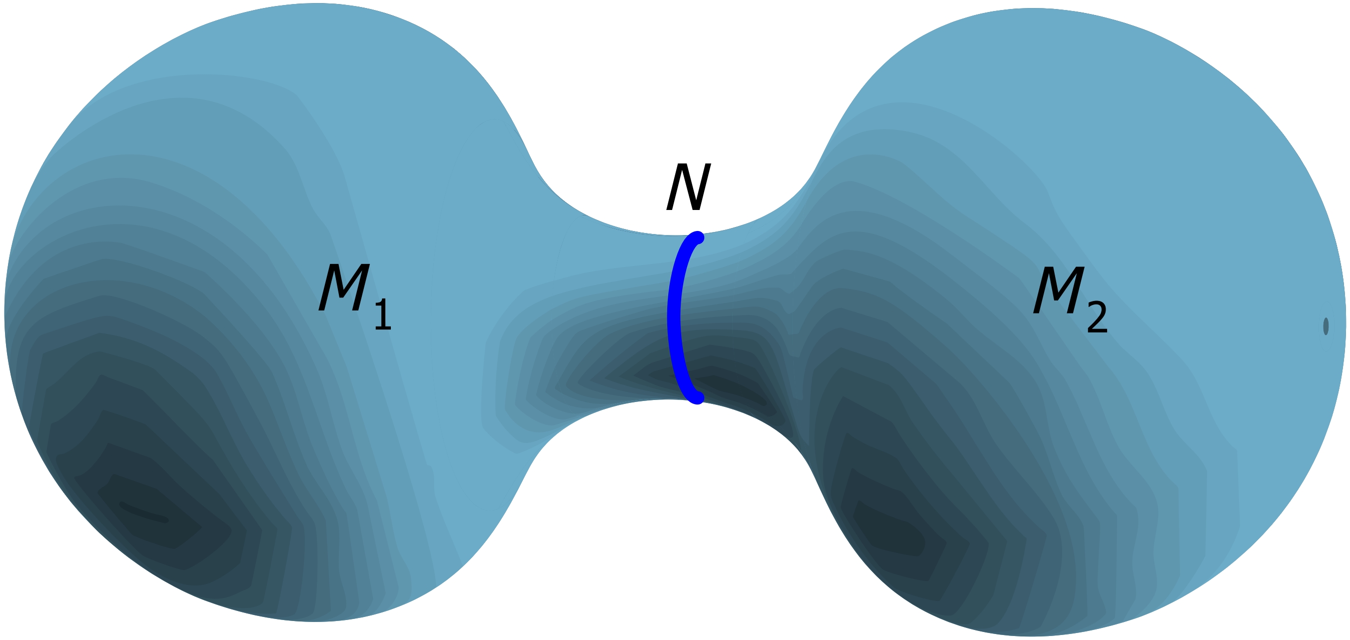
Harmonic forms and cohomology:

$$H^k(M;\mathbb{R}) \cong \{h \,|\, \Delta h = 0\}$$



Cheeger’s isoperimetric constant:

$$\lambda_C := \inf_N \left\{ \frac{\mathrm{vol}_{n-1}(N)}{\min(\mathrm{vol}_n(M_1),\, \mathrm{vol}_n(M_2))} \right\}$$



Cheeger's isoperimetric constant:

$$\lambda_C := \inf_N \left\{ \frac{\mathrm{vol}_{n-1}(N)}{\min(\mathrm{vol}_n(M_1),\ \mathrm{vol}_n(M_2))} \right\}$$

Cheeger-Buser:

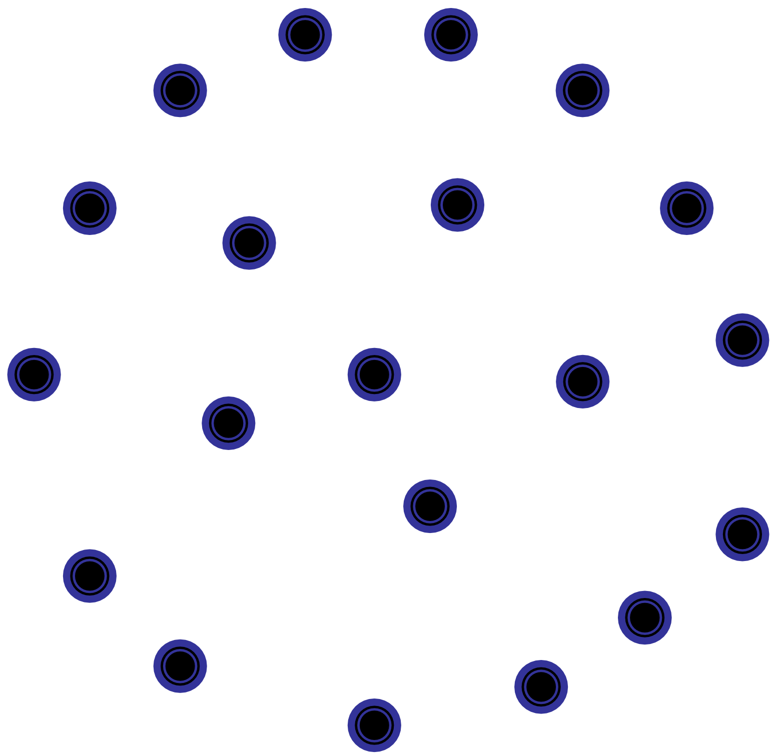
$$\frac{\lambda_C^2}{4} \leq \lambda_1 \leq c(K\lambda_C + \lambda_C^2)$$

1st non-trivial eigenvalue of Laplacian

3. Rippa’s theorem

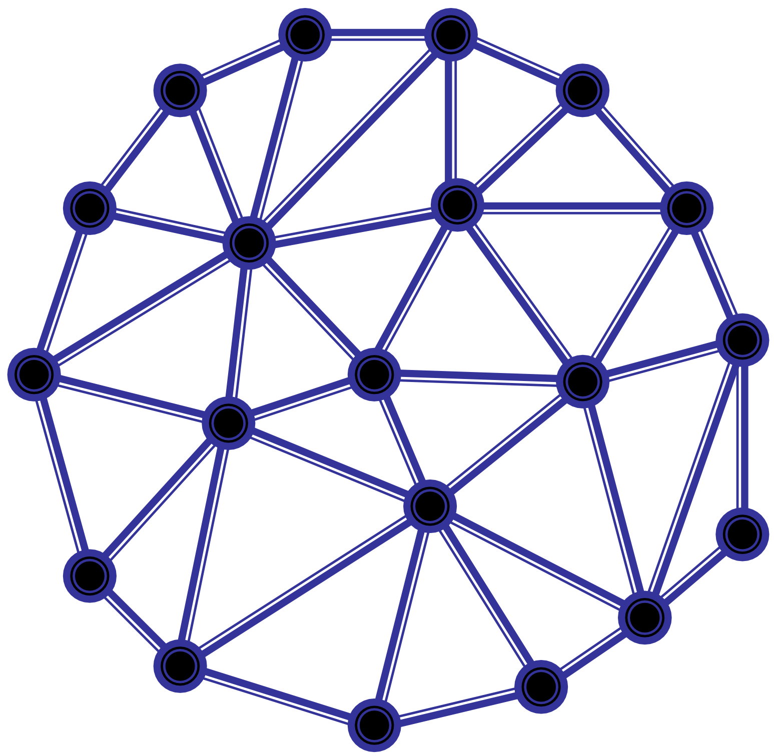
Consider

$u = (u_i)$ function on finite point set (in plane)



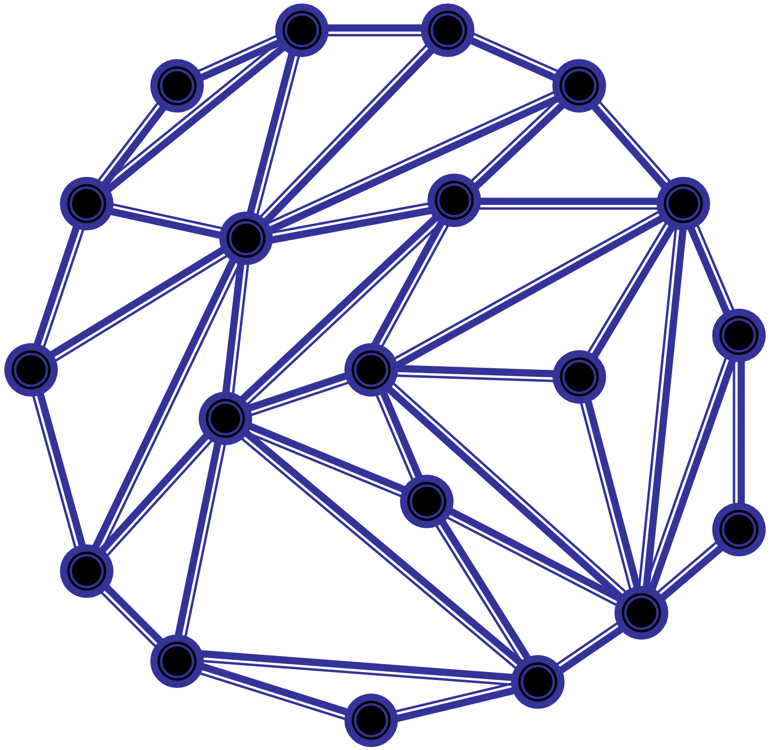
Consider

$u = (u_i)$ function on finite point set (in plane)



Consider

$u = (u_i)$ function on finite point set (in plane)

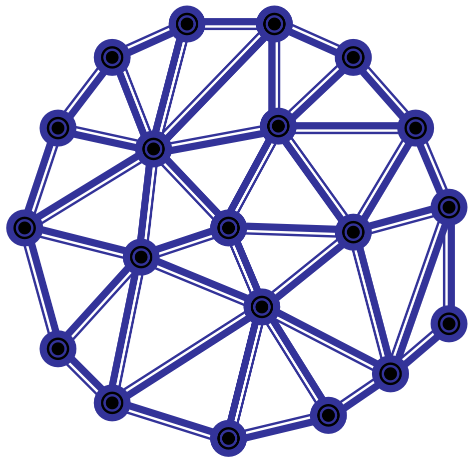


3. Rippa’s theorem

- Extend function piecewise linearly (hence continuous) over triangles.

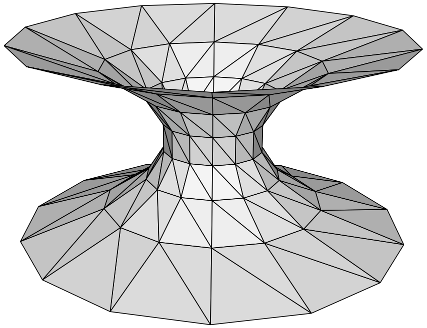
- Rippa’s theorem: Among all possible triangulations, the Delaunay triangulation minimizes the Dirichlet energy

$$E_D[u] := \frac{1}{2}\int_\Omega \nabla u \cdot \nabla u$$



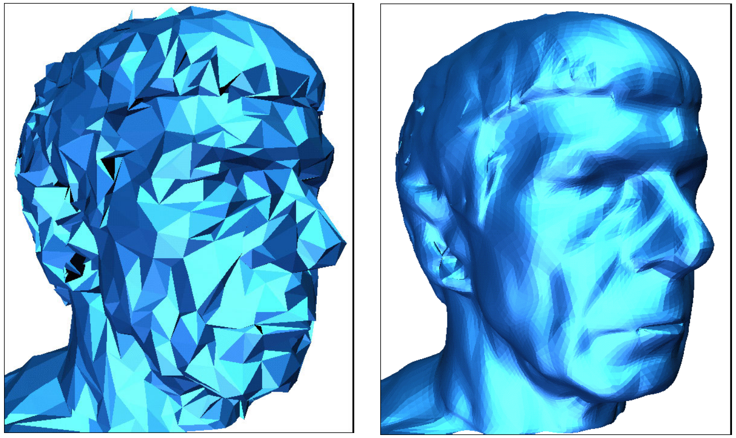
Laplacian on simplicial meshes

(mostly surfaces)



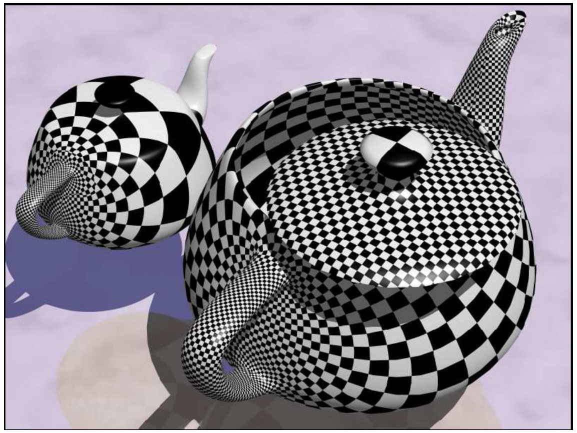
Discrete Laplacians – many geometric applications

mesh denoising



[Desbrun et al. '99]

parameterization

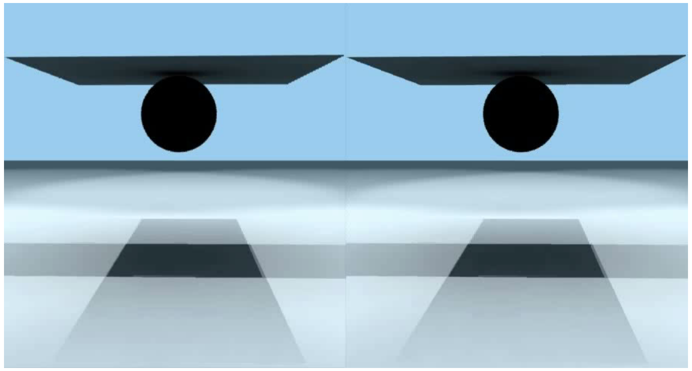


[Gu/Yau '03]

mesh editing

[Sorkine et al. '04]

simulation

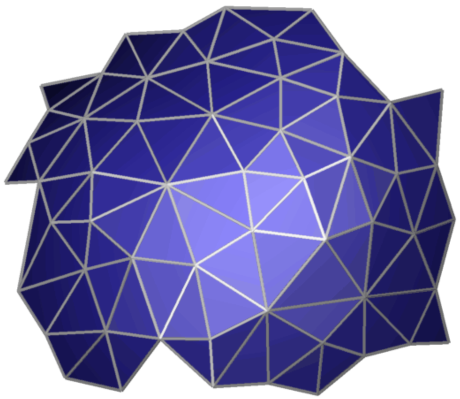


[Bergou et al. '06]

Key properties of smooth Laplacians (Sym) (Loc) (Lin) (Psd) (Max)

- (Sym) Symmetry: $(\Delta u, v)_{L^2} = (u, \Delta v)_{L^2}$
- (Loc) Locality: changing $u(q)$ does not change $(\Delta u)(p)$
- (Lin) Linear precision: $(\partial_x^2 + \partial_y^2)(ax + by + c) = 0$
- (Psd) Laplacians are positive (semi)definite
- (Max) Maximum principle

Discrete Laplace operators:



Input:

$u = (u_i)$ function on mesh vertices

Output:

$$(Lu)_i = \sum_j \omega_{ij}(u_i - u_j)$$

Properties of L are encoded by $\omega = (\omega_{ij})$

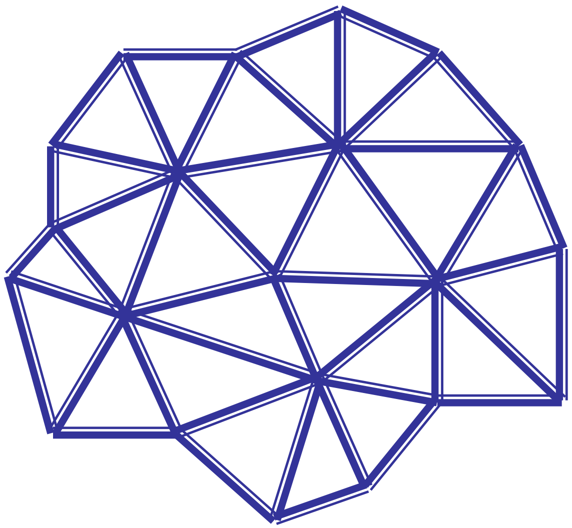
Desired properties

1. (Sym) Symmetry: $\omega_{ij} = \omega_{ji}$

Motivation:

- smooth symmetry
- real eigenvalues & orthogonal eigenvectors

2. (Loc) Locality: $\omega_{ij} = 0$ if (ij) is not an edge



Motivation:

- smooth locality
- diffusion: $u_t = -\Delta u$

  discrete: ω_{ij} random walk 'probabilities' along edges

Desired properties

3. (Lin) Linear precision: $(Lu)_i = 0$

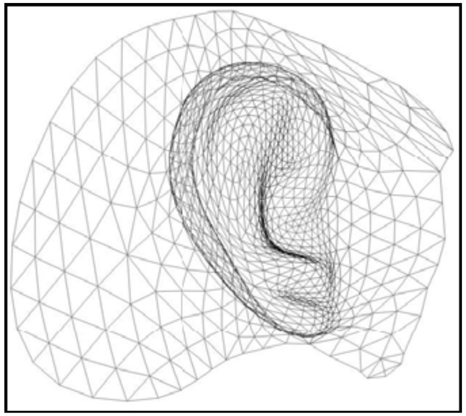
   if mesh is in the plane and u is linear

Motivation:

- smooth linear precision
- mesh denoising: no tangential vertex drift
- mesh parameterization: planar vertices don't move

4. (Pos) Positivity: $\omega_{ij} \geq 0$

$\Longrightarrow$ (Psd) + (Max)



[Gortler/Gotsman/Thurston ‘05]

Motivation:

- positive (semi)-definiteness
- parameterization: no flipped triangles (locally)
- barycentric coordinates (maximum principle)

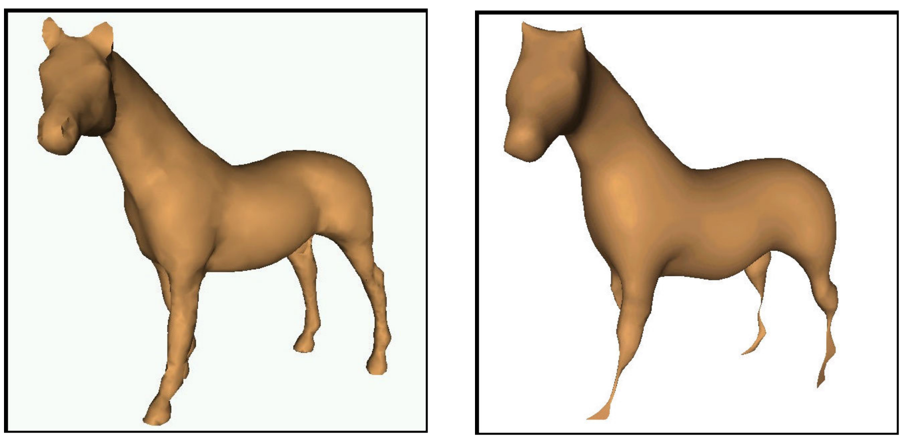
$$\lambda_{ij} = \frac{\omega_{ij}}{\sum_{j \neq i} \omega_{ij}} \quad \Longrightarrow \quad \sum_{j \neq i} \lambda_{ij} = 1$$

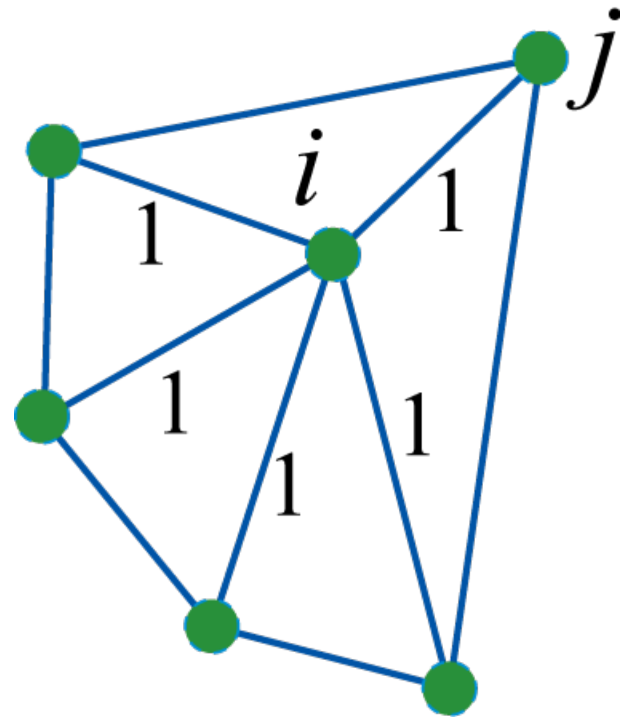
Four examples

(Sym) (Loc) (Lin) (Pos)

1. Combinatorial Laplacians [Tutte ’63, …]

$$\omega_{ij} = 1$$





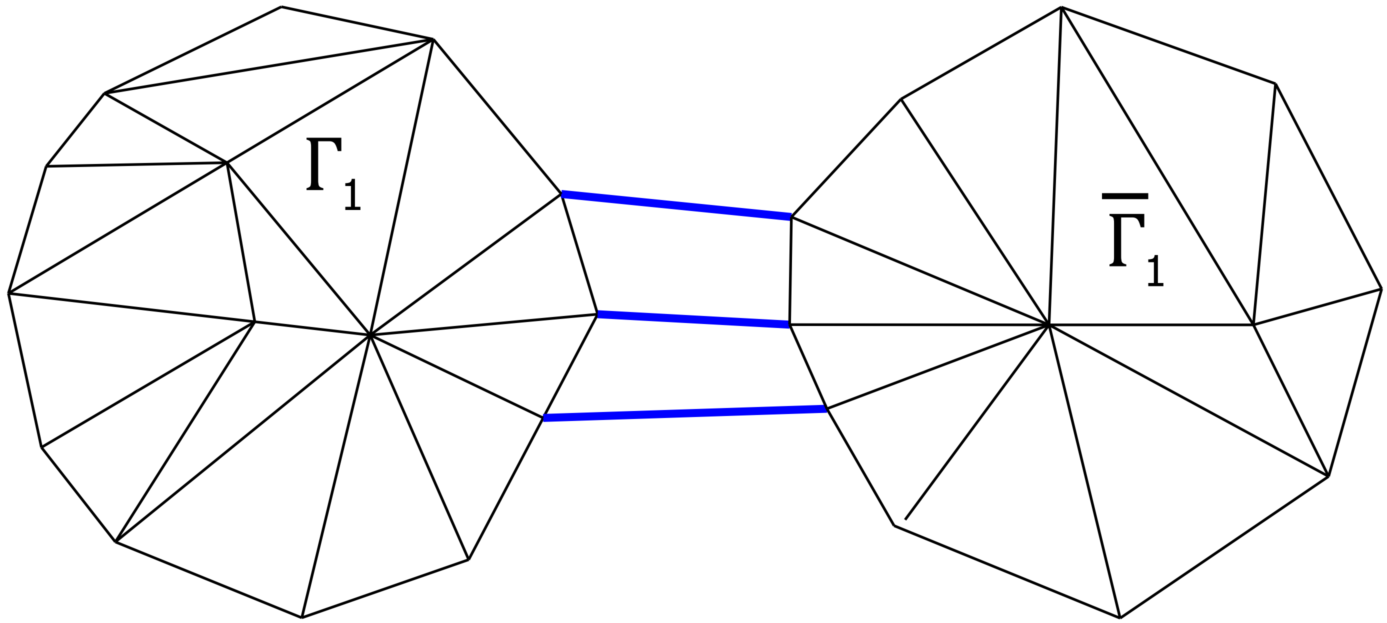
[Karni/Gotsman ’00]

Graph Laplacian cont.

(Sym) (Loc) (Lin) (Pos)

Cheeger constant: $$\lambda_C := \min\left\{\frac{\mathrm{vol}(\Gamma_1, \bar{\Gamma}_1)}{\min(\mathrm{vol}(\Gamma_1),\ \mathrm{vol}(\bar{\Gamma}_1))}\right\}$$

$$\mathrm{vol}(\Gamma_1, \bar{\Gamma}_1) := \sum_{i \in V_1, j \notin V_1} \omega_{ij} \quad \text{and} \quad \mathrm{vol}(\Gamma_1) := \sum_{i \in V_1, j \in V_1} \omega_{ij}$$



Graph Laplacian cont.

(Sym) (Loc) (Lin) (Pos)

Cheeger constant: $\lambda_C := \min \left\{ \frac{\mathrm{vol}(\Gamma_1, \bar{\Gamma}_1)}{\min(\mathrm{vol}(\Gamma_1),\ \mathrm{vol}(\bar{\Gamma}_1))} \right\}$

$$\mathrm{vol}(\Gamma_1, \bar{\Gamma}_1) := \sum_{i \in V_1, j \notin V_1} \omega_{ij} \quad \text{and} \quad \mathrm{vol}(\Gamma_1) := \sum_{i \in V_1, j \in V_1} \omega_{ij}$$

Discrete Cheeger inequality:

$$\frac{\lambda_C^2}{2} \le \tilde{\lambda}_1 \le 2\lambda_C$$

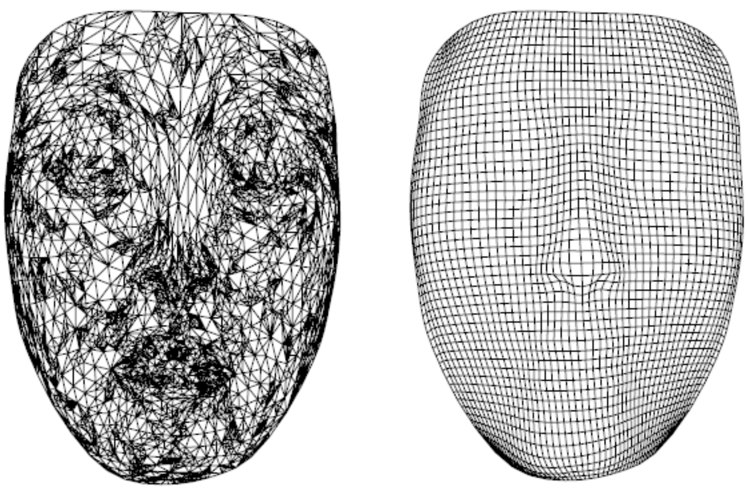
$\tilde{\lambda}_1$: 1st non-trivial eigenvalue of (normalized) graph Laplacian

Four examples

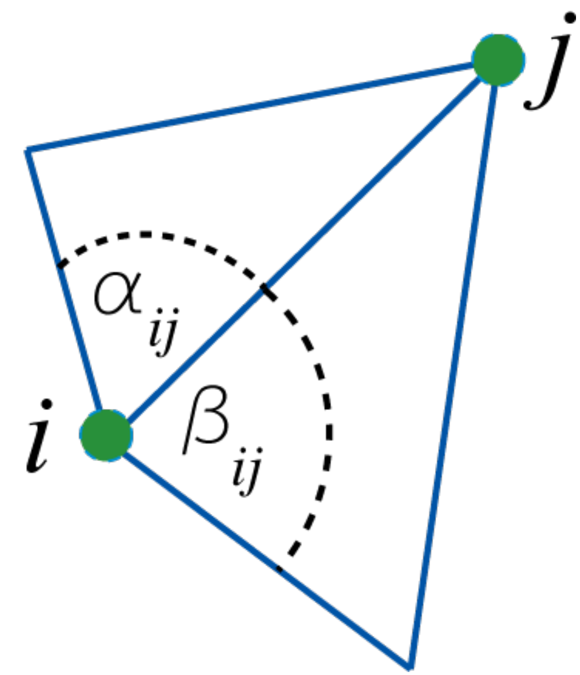
(Sym) (Loc) (Lin) (Pos)

2. Mean-value coordinates [Floater '03, …]

$$\omega_{ij} = \frac{\tan\frac{\alpha_{ij}}{2} + \tan\frac{\beta_{ij}}{2}}{|e_{ij}|}$$



[Floater '03]



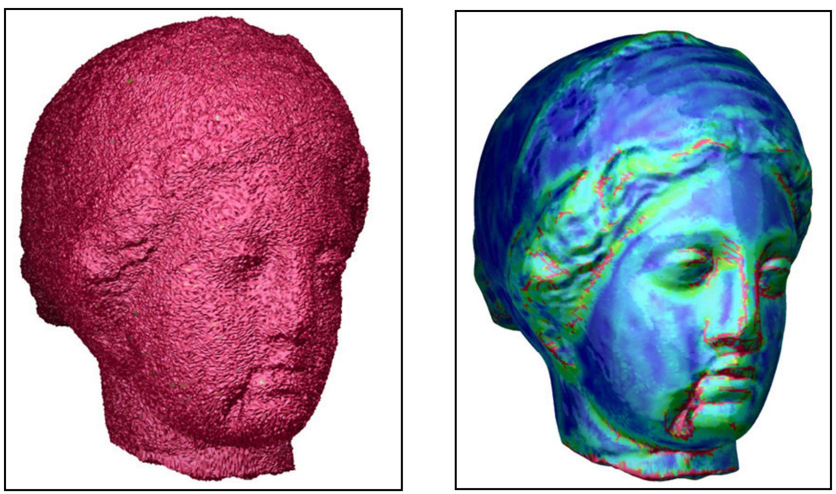
Four examples

(Sym) (Loc) (Lin) (Pos)

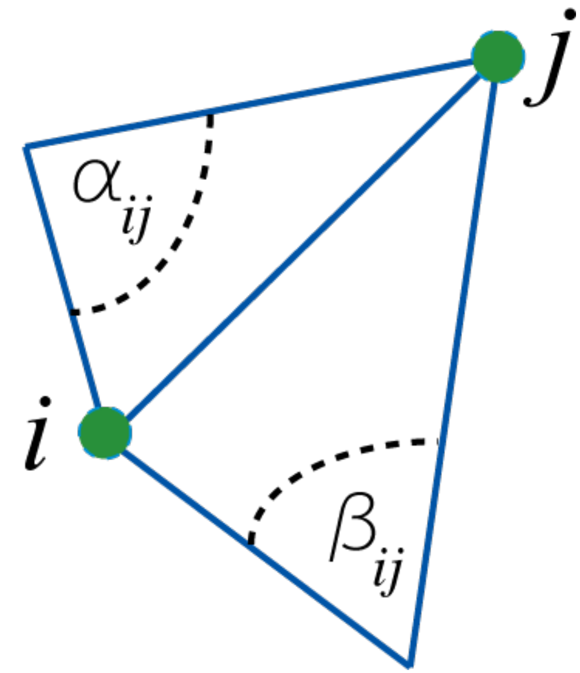
3. cotan weights [Pinkall/Poltier ’93, McNeal ’49, …]

$$\omega_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}$$

$$\alpha_{ij} + \beta_{ij} > \pi \iff \omega_{ij} < 0$$



[Hildebrandt/Polthier ’05]



cotan Laplacian cont.

(Sym) (Loc) (Lin) (Pos)

C^k = simplicial cochains (dual to simplicial k-chains)

C^0 : real values at vertices
C^1 : dual to oriented edges
C^2 : dual to oriented triangles
...

simplicial coboundary operator: $\delta : C^k \rightarrow C^{k+1}$

inner product on k-cochains: $(\alpha, \beta)_k$

simplicial codifferential: $(\alpha, \delta^* \beta)_k = (\delta\alpha, \beta)_{k+1}$

$$\mathbb{L} := \delta^* \delta + \delta \delta^*$$

cotan Laplacian cont.

(Sym) (Loc) (Lin) (Pos)

$$\mathbb{L} := \delta^* \delta + \delta \delta^*$$

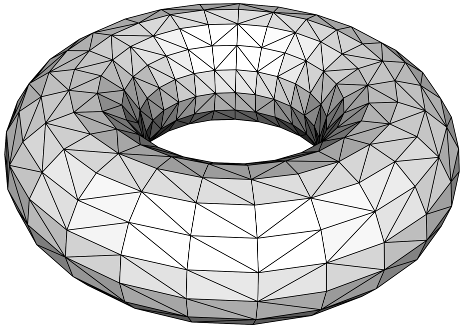
Hodge-Helmholtz decomposition of k-cochains:

$$\alpha = \delta\mu + \delta^*\nu + h$$

$$\mathbb{L}h = 0$$

Harmonic forms and cohomology:

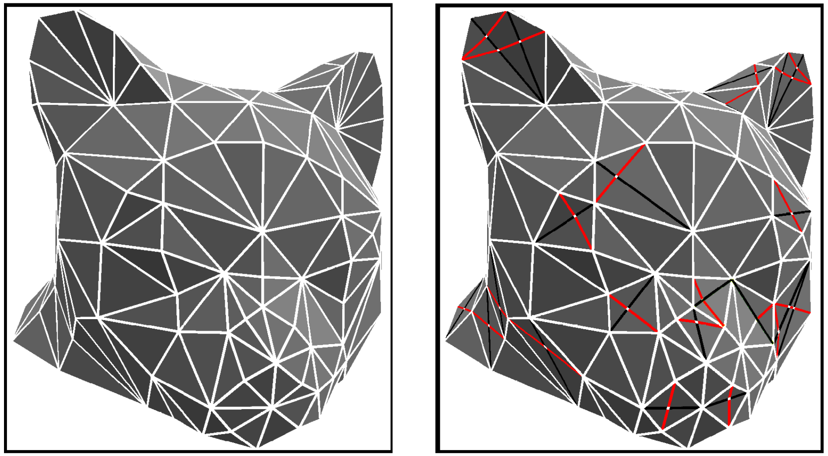
$$H^k(M;\mathbb{R}) \cong \{h \,|\, \mathbb{L}h = 0\}$$



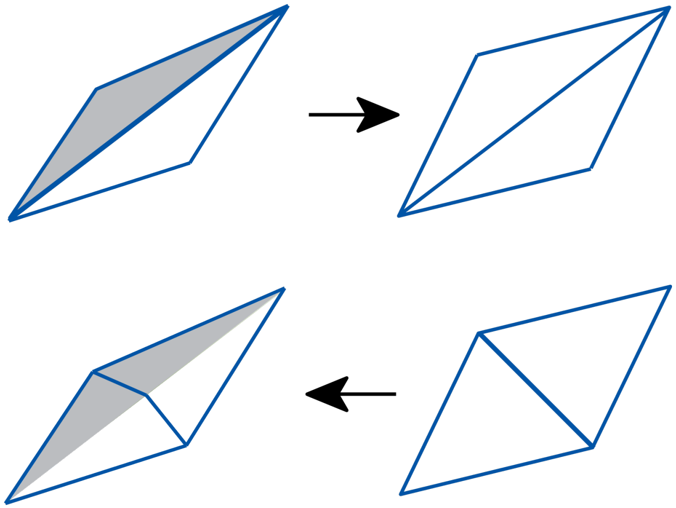
- Whitney-map (lin. interpolation) & L^2 inner product lead to cotan Laplacian

4. intrinsic Delaunay [Bobenko/Sprinborn '05, ...]

$$\omega_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}$$
$$\geq 0$$



[Fisher et al. '06]



[intrinsic edge flips]

Putting four things together

	(Sym)	(Loc)	(Lin)	(Pos)
mean value	ø	✓	✓	✓
intrinsic Delaunay	✓	ø	✓	✓
combinatorial	✓	✓	ø	✓
cotan	✓	✓	✓	ø

… on general irregular meshes!

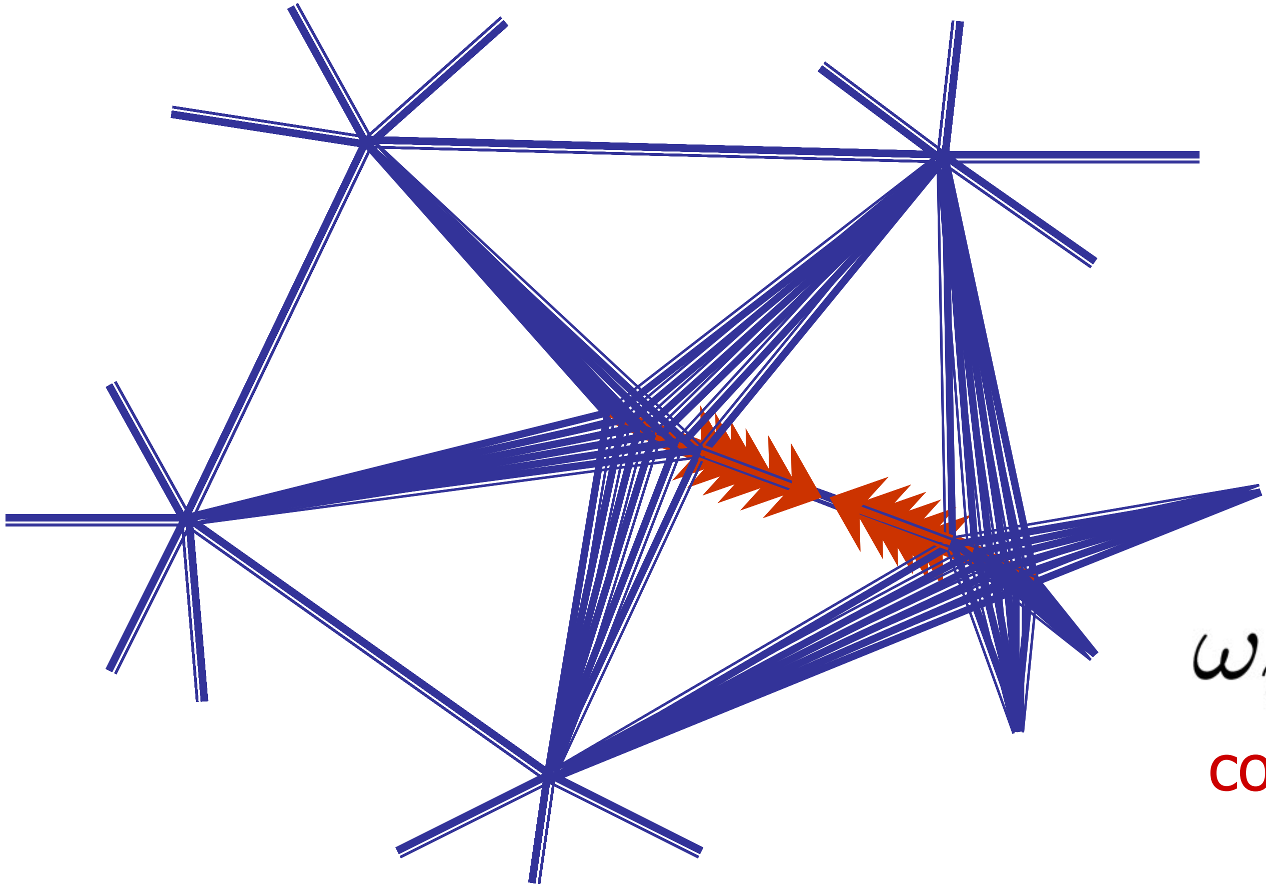
Putting four things together

No-free-lunch-theorem (preliminary version)

General meshes do not allow for discrete Laplacians with (Sym)+(Loc)+(Lin)+(Pos).

Sketch of proof

1. (Sym)+(Loc) & stress frameworks in the plane

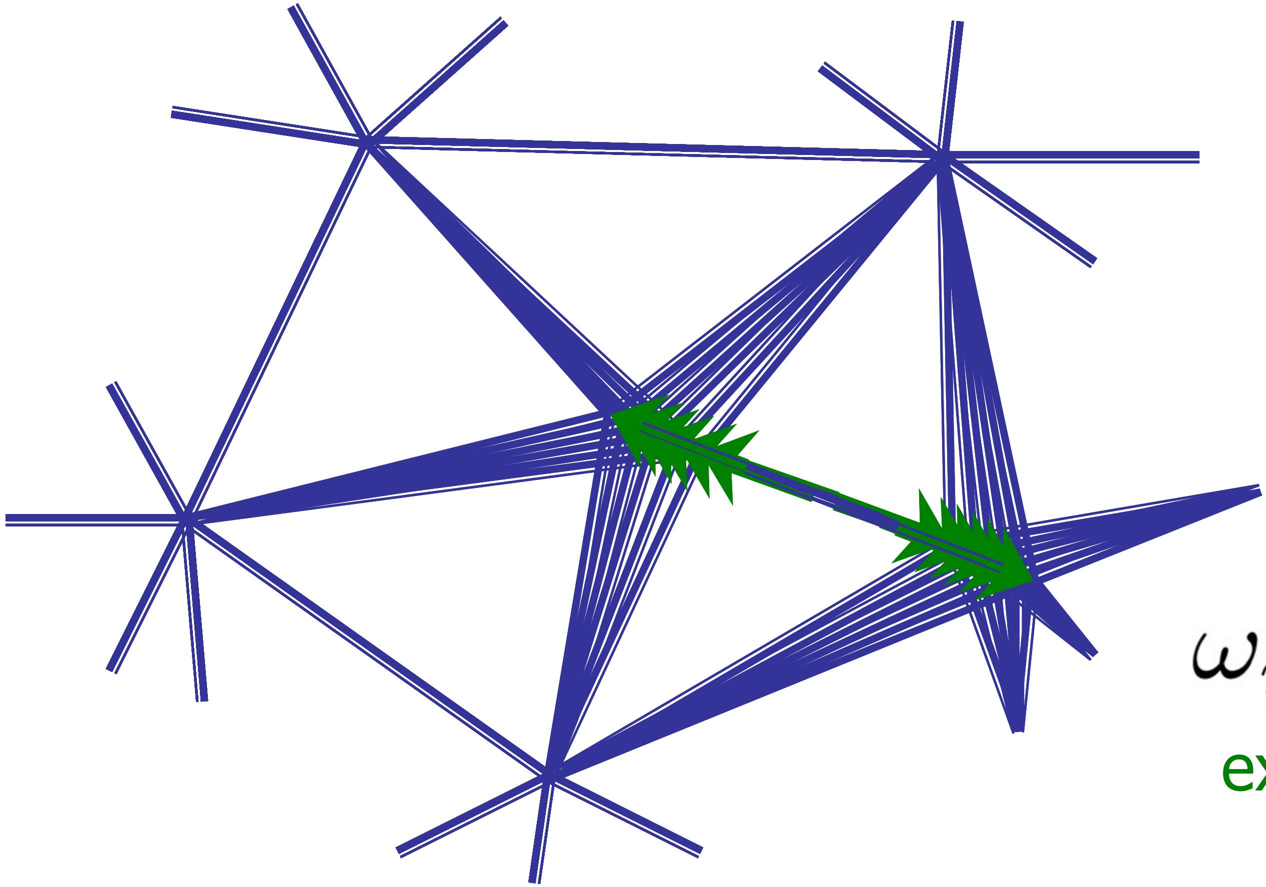


$\omega_{ij} > 0$

contracting

Proof: planar stress frameworks

1. (Sym)+(Loc) & stress frameworks in the plane



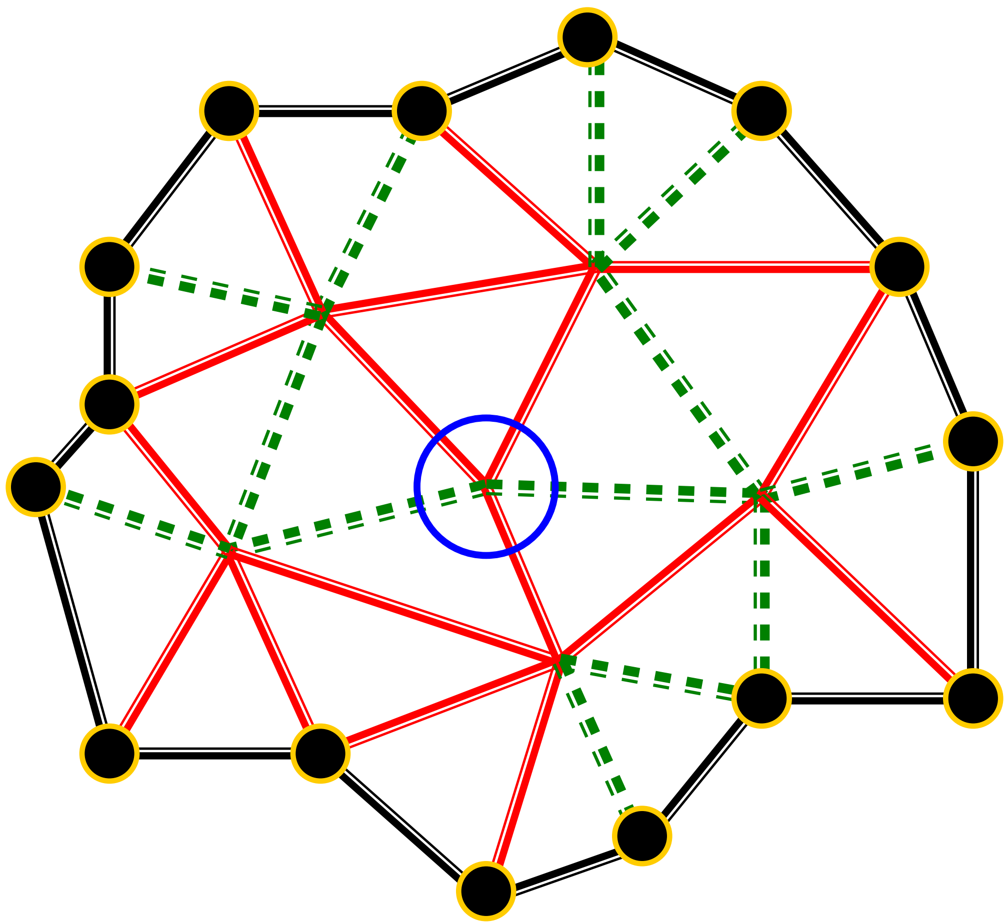
$\omega_{ij} < 0$

expanding

Proof: planar stress frameworks

2. (Sym)+(Loc)+(Lin) $\Longleftrightarrow$ **inner vertices are in force balance**

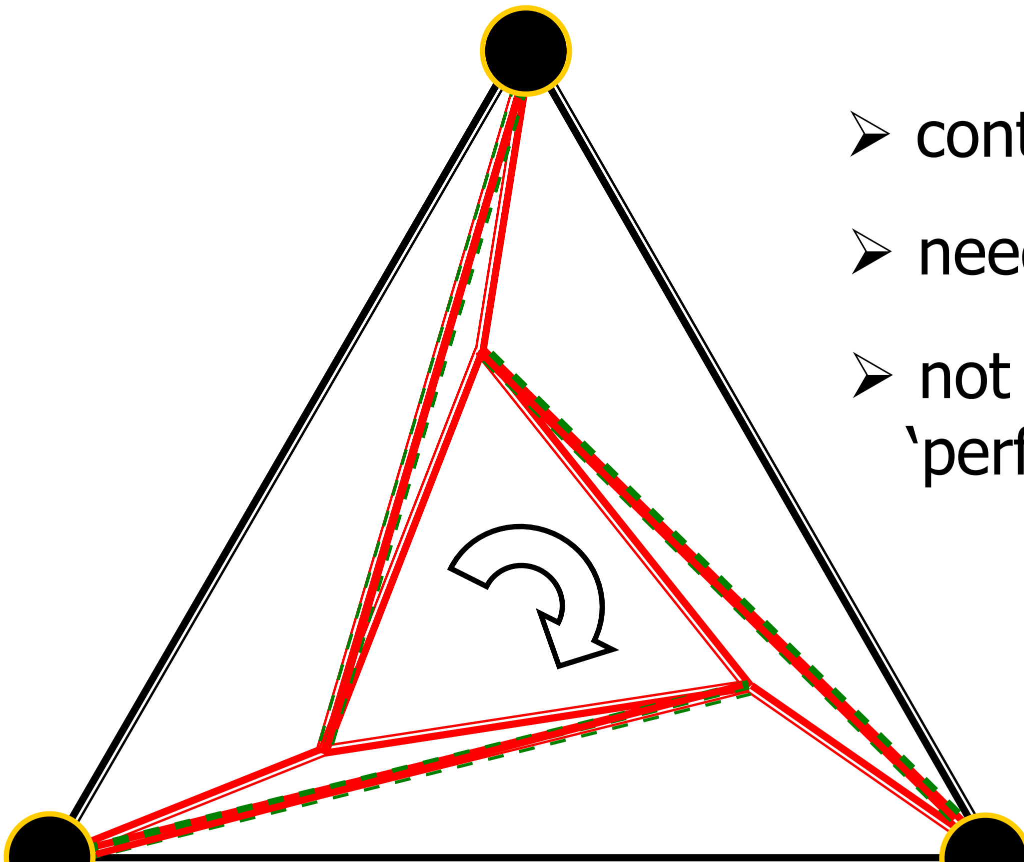
[e.g., use cotan weights]



- fixed boundary vertices
- contracting edges
- expanding edges

Proof: planar stress frameworks

3. (Sym)+(Loc)+(Lin)+(Pos)



[Schönhardt polytope]

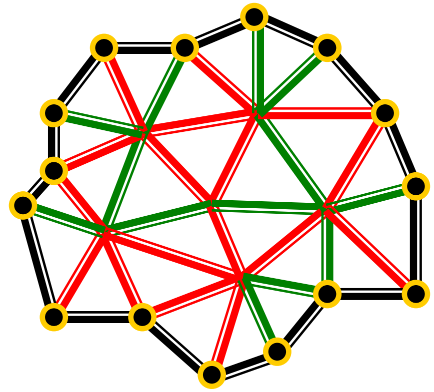
- contracting forces: net torque
- need negative weights
- not all meshes allow for 'perfect' Laplacians

"QED"

Which meshes allow for

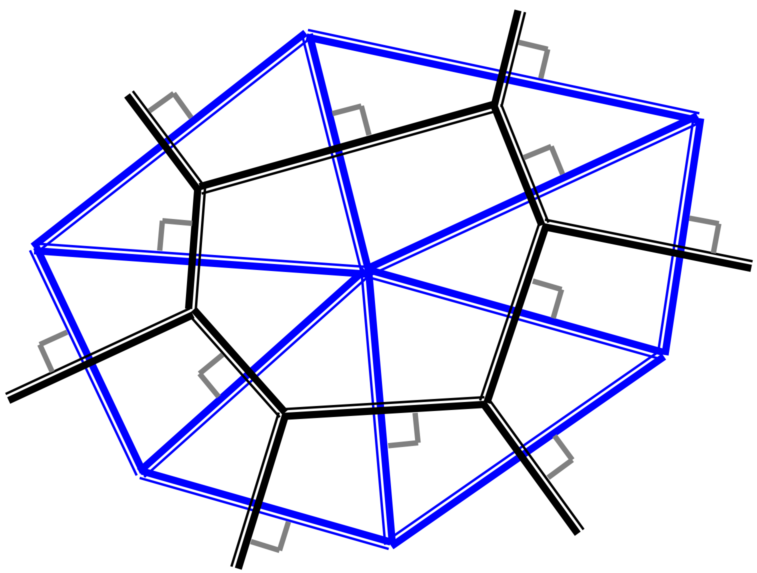
‘perfect’ Laplacians?

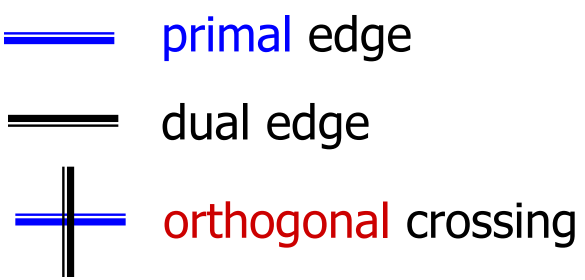
Which meshes allow for ‘perfect’ Laplacians?



Theorem (Maxwell-Cremona 1864)

A stress framework in the plane is in force-balance iff there exists an orthogonal dual graph.





Example: Delaunay triangulation & Voronoi dual

Which meshes allow for 'perfect' Laplacians?

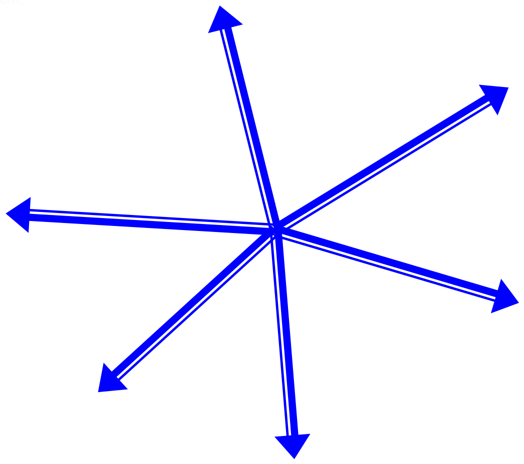
Theorem (Maxwell-Cremona 1864)

(Sym)+(Loc)+(Lin) $\Longleftrightarrow$ orthogonal duals

Proof:

1) Given (Sym)+(Loc)+(Lin), observe that

$$\sum_j \omega_{ij}\vec{e}_{ij} = \sum_j \omega_{ij}(p_j - p_i) = 0$$



Which meshes allow for 'perfect' Laplacians?

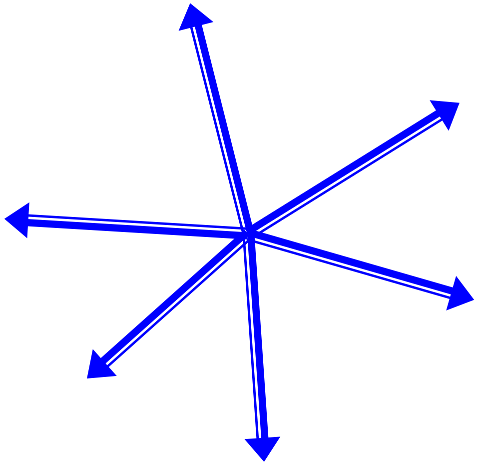
Theorem (Maxwell-Cremona 1864)

(Sym)+(Loc)+(Lin) $\Longleftrightarrow$ orthogonal duals

Proof:

1) Given (Sym)+(Loc)+(Lin), observe that

$$\sum_j \omega_{ij}\vec{e}_{ij} = \sum_j \omega_{ij}(p_j - p_i) = 0$$



Which meshes allow for 'perfect' Laplacians?

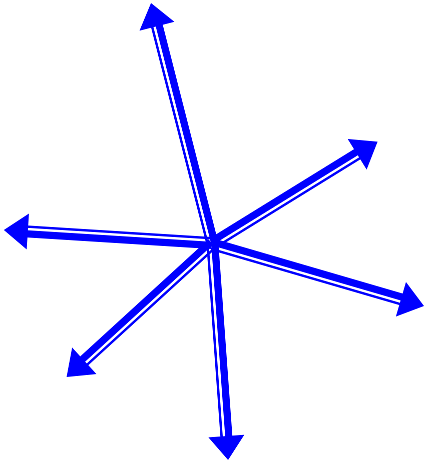
Theorem (Maxwell-Cremona 1864)

(Sym)+(Loc)+(Lin) $\Longleftrightarrow$ orthogonal duals

Proof:

1) Given (Sym)+(Loc)+(Lin), observe that

$$\sum_j \omega_{ij} \vec{e}_{ij} = \sum_j \omega_{ij}(p_j - p_i) = 0$$



Which meshes allow for ‘perfect’ Laplacians?

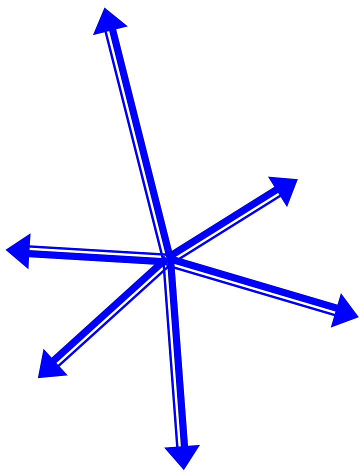
Theorem (Maxwell-Cremona 1864)

(Sym)+(Loc)+(Lin) $\Longleftrightarrow$ orthogonal duals

Proof:

1) Given (Sym)+(Loc)+(Lin), observe that

$$\sum_j \omega_{ij}\vec{e}_{ij} = \sum_j \omega_{ij}(p_j - p_i) = 0$$



Which meshes allow for ‘perfect’ Laplacians?

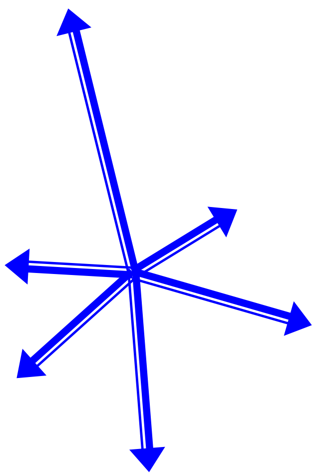
Theorem (Maxwell-Cremona 1864)

(Sym)+(Loc)+(Lin) $\Longleftrightarrow$ orthogonal duals

Proof:

1) Given (Sym)+(Loc)+(Lin), observe that

$$\sum_j \omega_{ij}\vec{e}_{ij} = \sum_j \omega_{ij}(p_j - p_i) = 0$$



Which meshes allow for 'perfect' Laplacians?

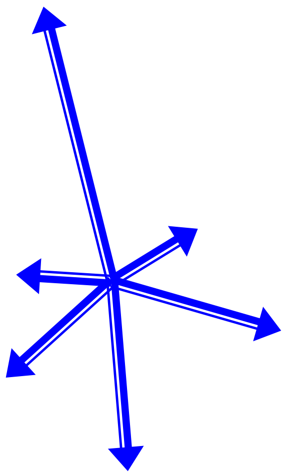
Theorem (Maxwell-Cremona 1864)

(Sym)+(Loc)+(Lin) $\Longleftrightarrow$ orthogonal duals

Proof:

1) Given (Sym)+(Loc)+(Lin), observe that

$$\sum_j \omega_{ij}\vec{e}_{ij} = \sum_j \omega_{ij}(p_j - p_i) = 0$$



Which meshes allow for 'perfect' Laplacians?

Theorem (Maxwell-Cremona 1864)

(Sym)+(Loc)+(Lin) $\Longleftrightarrow$ orthogonal duals

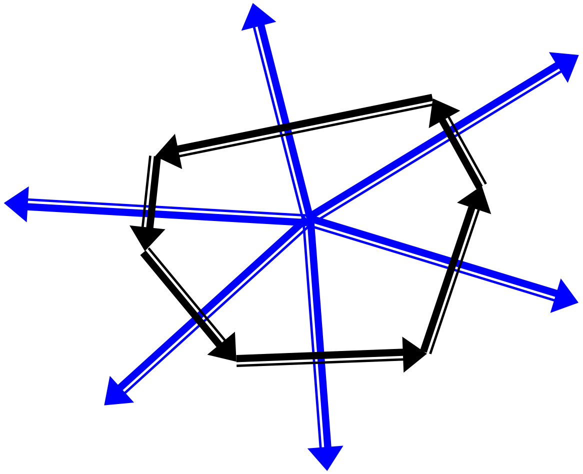
Proof:

1) Define dual edges by

$$\star\vec{e}_{ij} = R^{90}(\omega_{ij}\vec{e}_{ij})$$

Get closed dual cycles.

$$\sum_j \star\vec{e}_{ij} = 0$$



Which meshes allow for ‘perfect’ Laplacians?

Theorem (Maxwell-Cremona 1864)

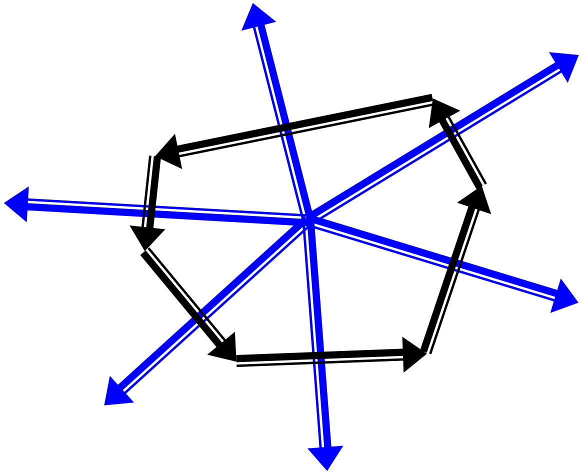
(Sym)+(Loc)+(Lin) $\Longleftrightarrow$ orthogonal duals

Proof:

2) Vice-versa, given orthogonal dual, define

$$\omega_{ij} = \frac{|\star e_{ij}|}{|e_{ij}|}$$

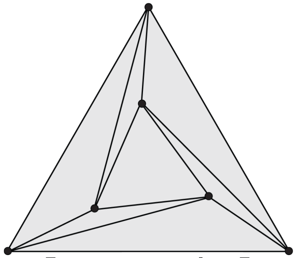
Closed dual cycles give (Lin).



Which meshes allow for 'perfect' Laplacians?

Theorem (Maxwell-Cremona 1864)

(Sym)+(Loc)+(Lin) $\Longleftrightarrow$ orthogonal duals


[not regular]

&

Theorem (Aurenhammer 1987)

Orthogonal duals w/ pos. weights $\Longleftrightarrow$ regular triangulations

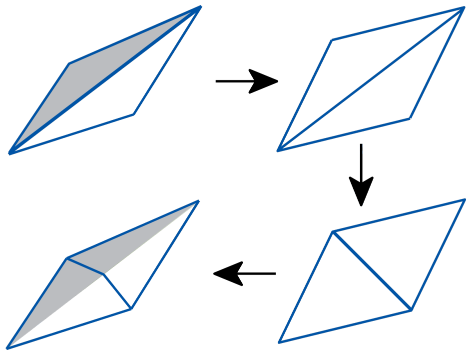
No-free-lunch-theorem (W., Mathur, Kälberer, Grinspun)

(Sym)+(Loc)+(Lin)+(Pos) $\Longleftrightarrow$ regular triangulations

Regular triangulations

Regular triangulations:

- ➢ Delaunay
- ➢ more generally: weighted Delaunay



[Edelsbrunner/Shah ’92,
Bobenko/Springborn ’05,
Glickenstein ’05]

Intrinsic weighted-Delaunay-Laplacians:

- ➢ break (Loc) wrt. input mesh

Taxonomy on literature

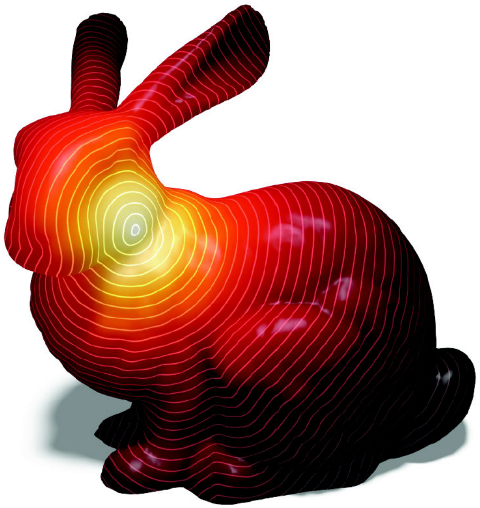
Laplacian Zoo

- dropping (Loc): weighted Delaunay Laplacians
- dropping (Sym): barycentric coordiantes
- dropping (Lin): combinatorial Laplacians
- dropping (Pos): cotan weights and generalizations

… no free lunch!

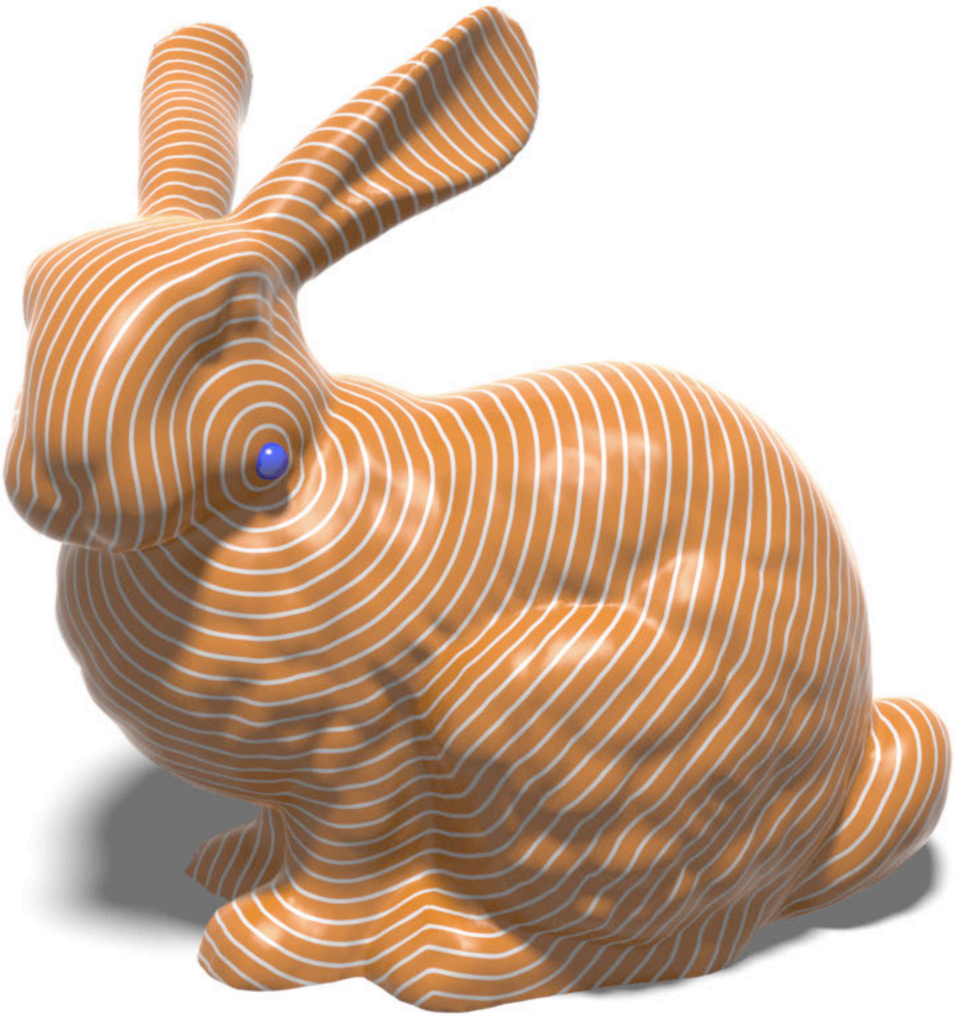
Application:

Geodesic distance computation

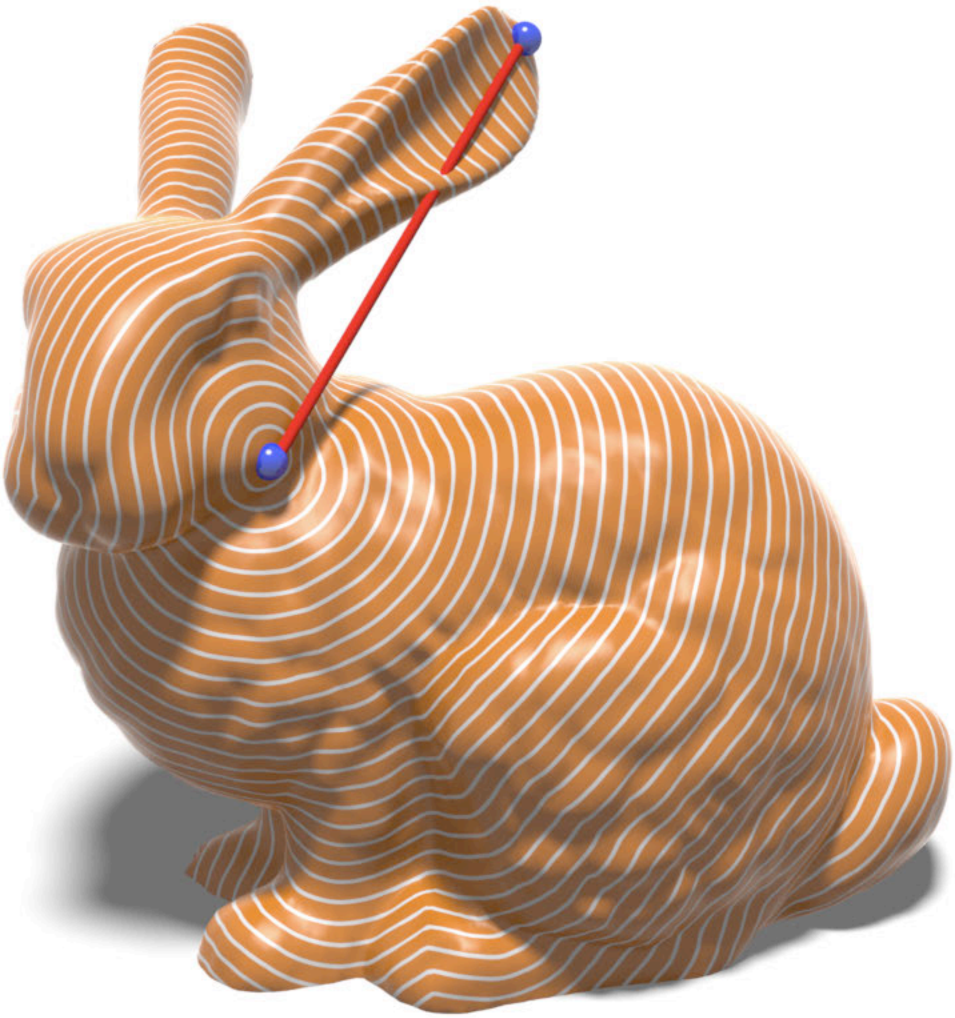


Problem formulation

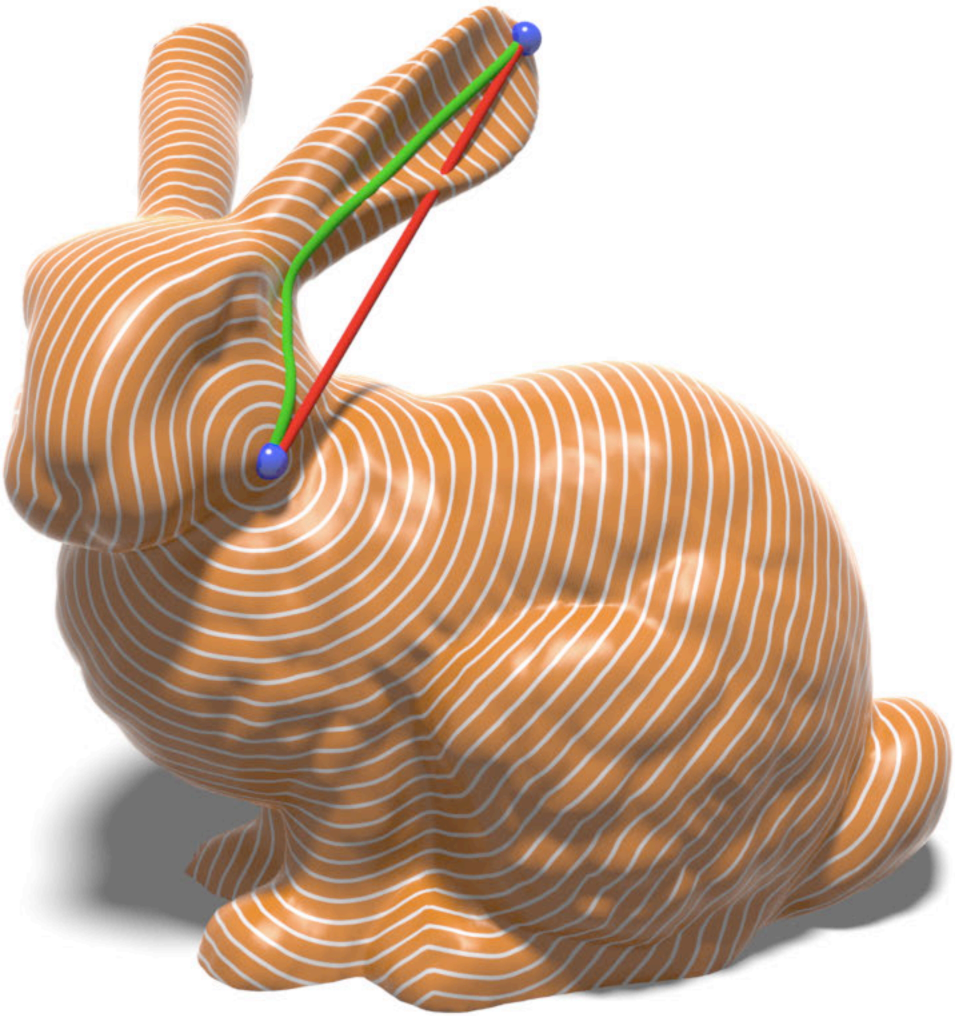
Problem formulation



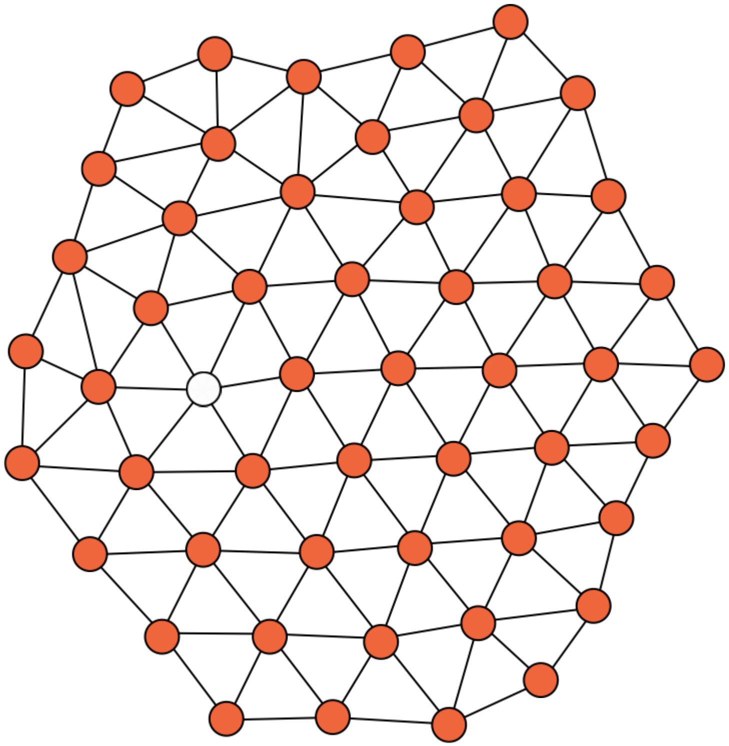
Problem formulation

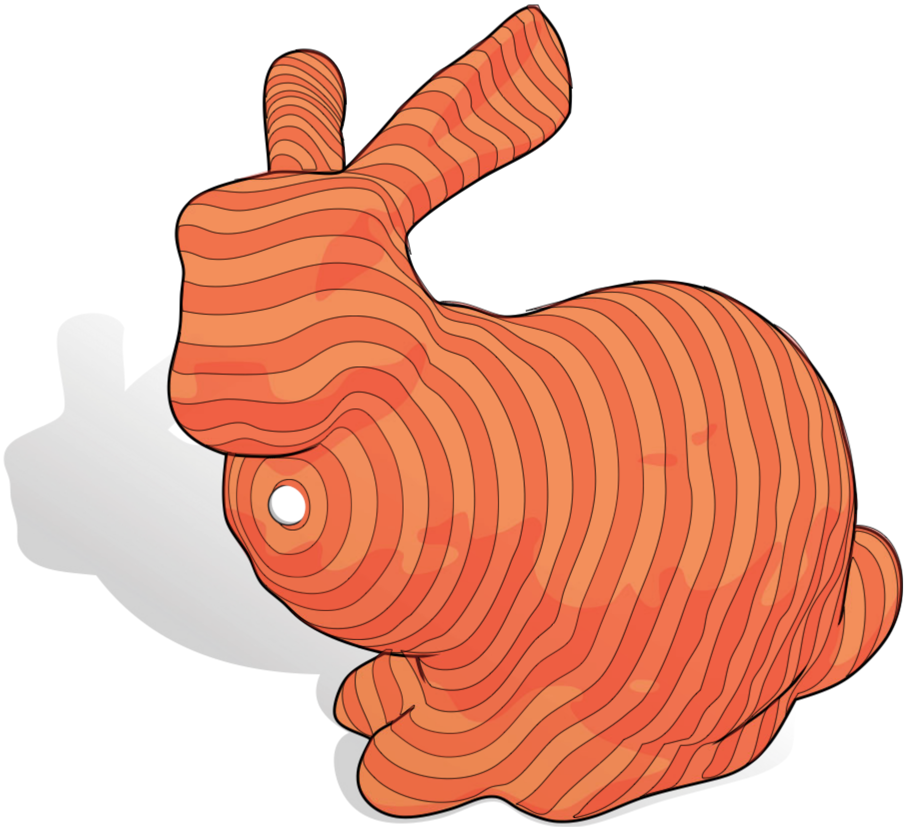


Problem formulation



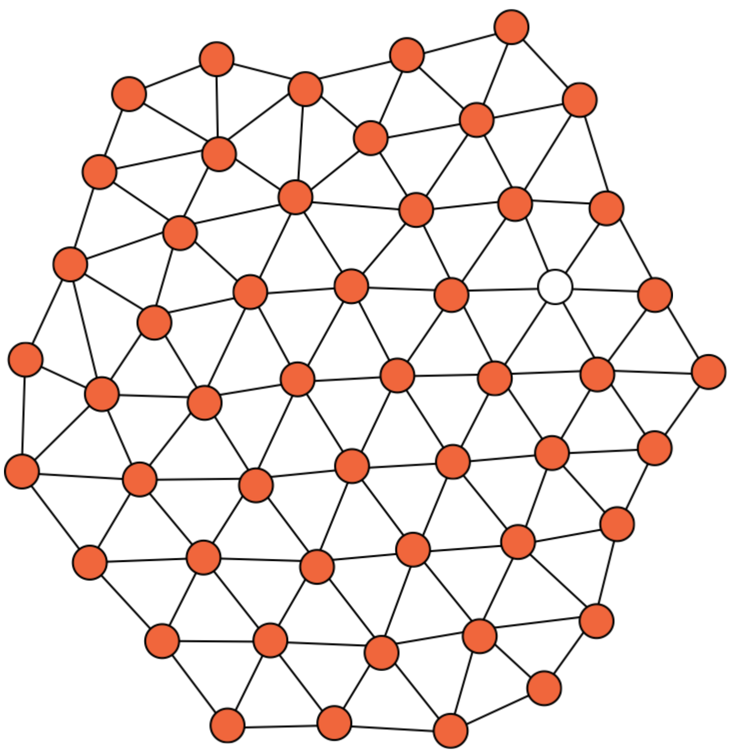
Challenges

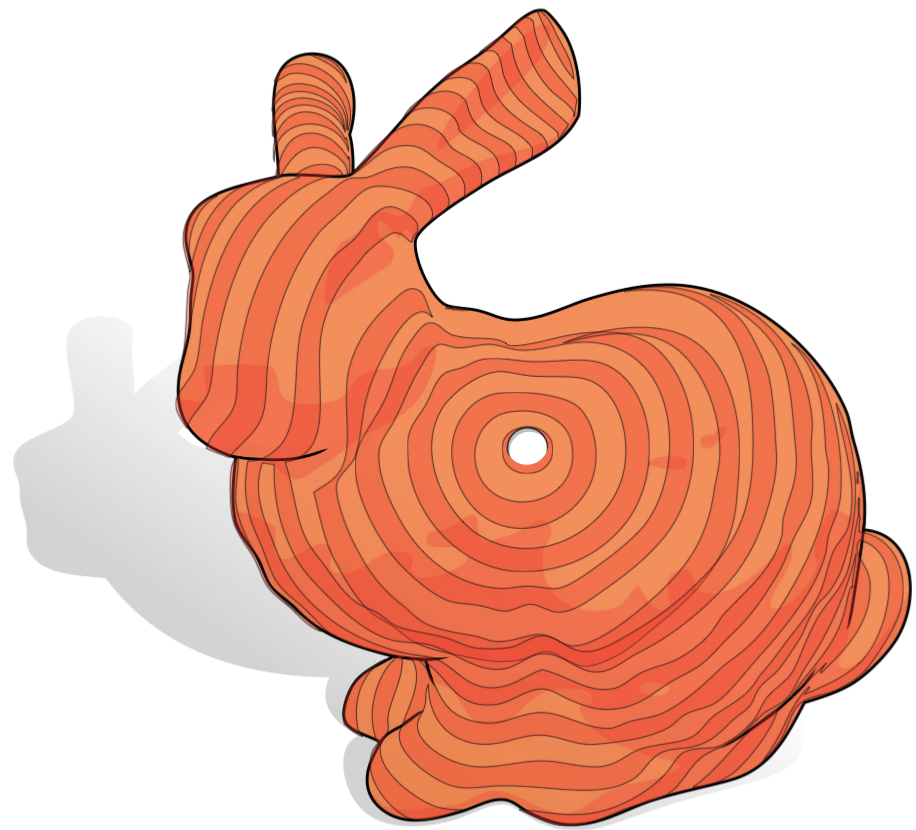




[Dijkstra 1959, Mitchell et al 1987, Chen & Han 1990, Sethian & Kimmel 1998, Surazhsky et al 2005...]

Challenges

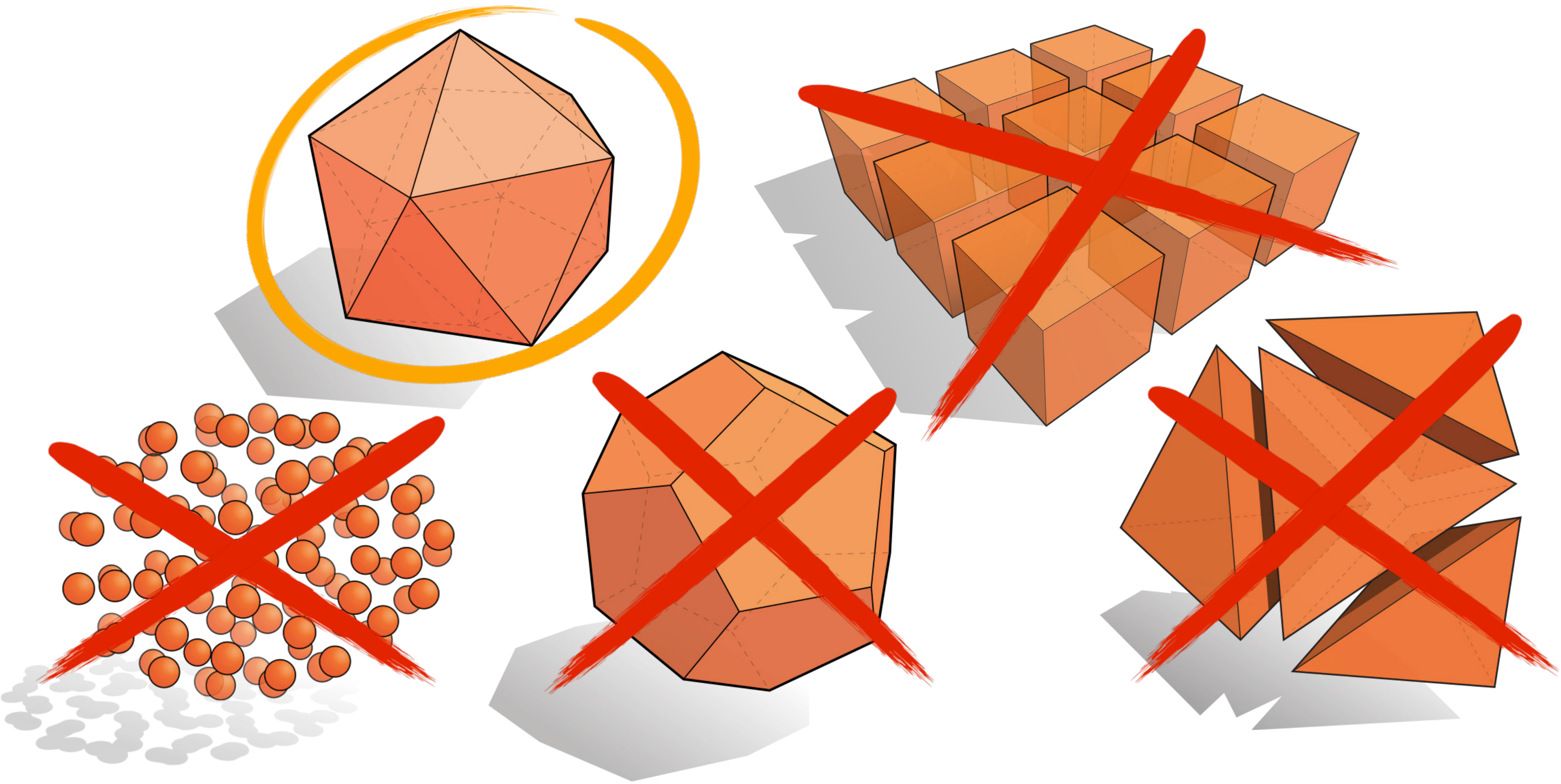




[Dijkstra 1959, Mitchell et al 1987, Chen & Han 1990, Sethian & Kimmel 1998, Surazhsky et al 2005...]

Challenges

Challenges



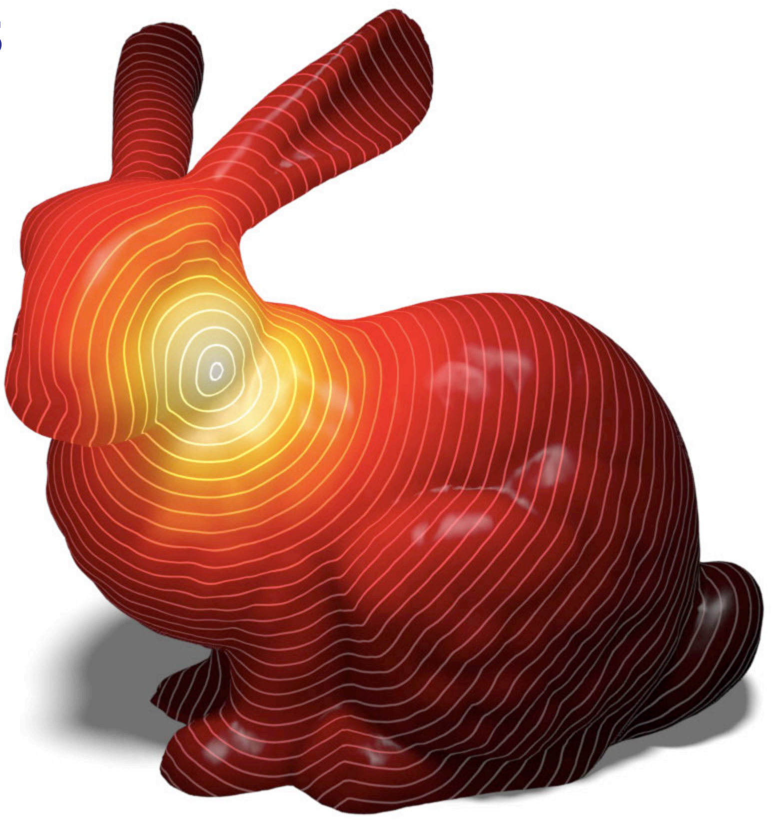
The Heat Method [Crane, Weischedel, W. 2013]

Solve two standard linear equations

- heat equation
- Poisson equation

Fast, general, simple

- parallelize
- prefactor
- any spatial discretization
- easy to implement

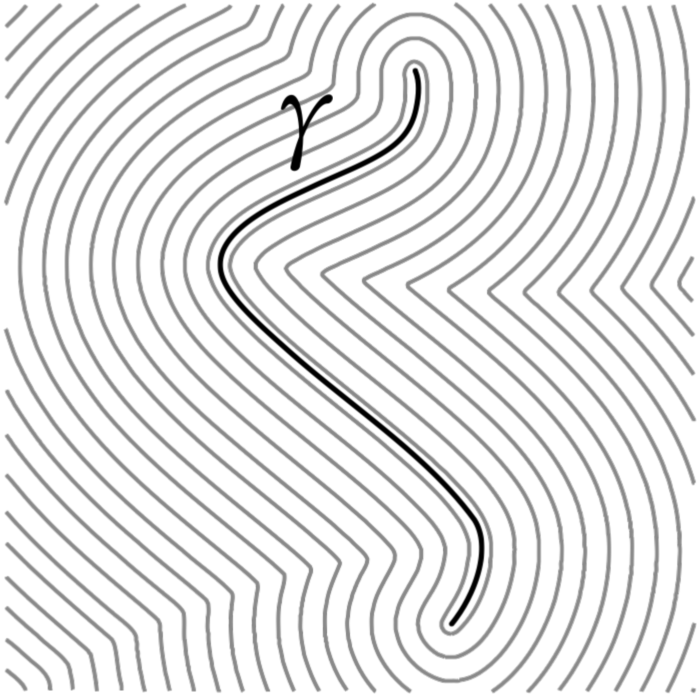


Eikonal equation

distance to source

$$|\nabla \phi| = 1$$

"distance changes at one meter per meter"



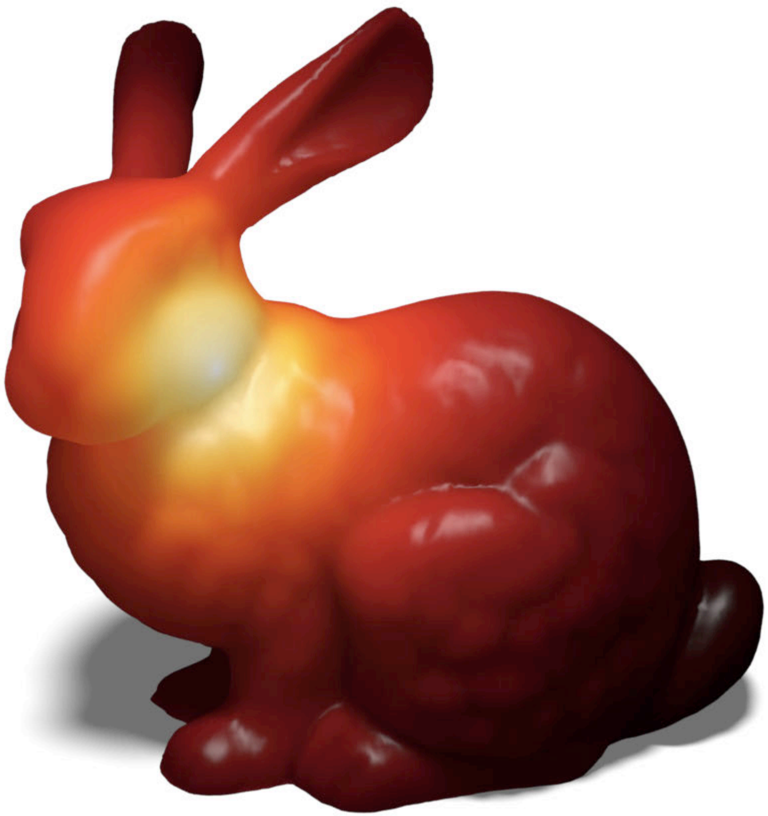
$$\phi|_\gamma = 0$$

Varadhan's formula

distance to source

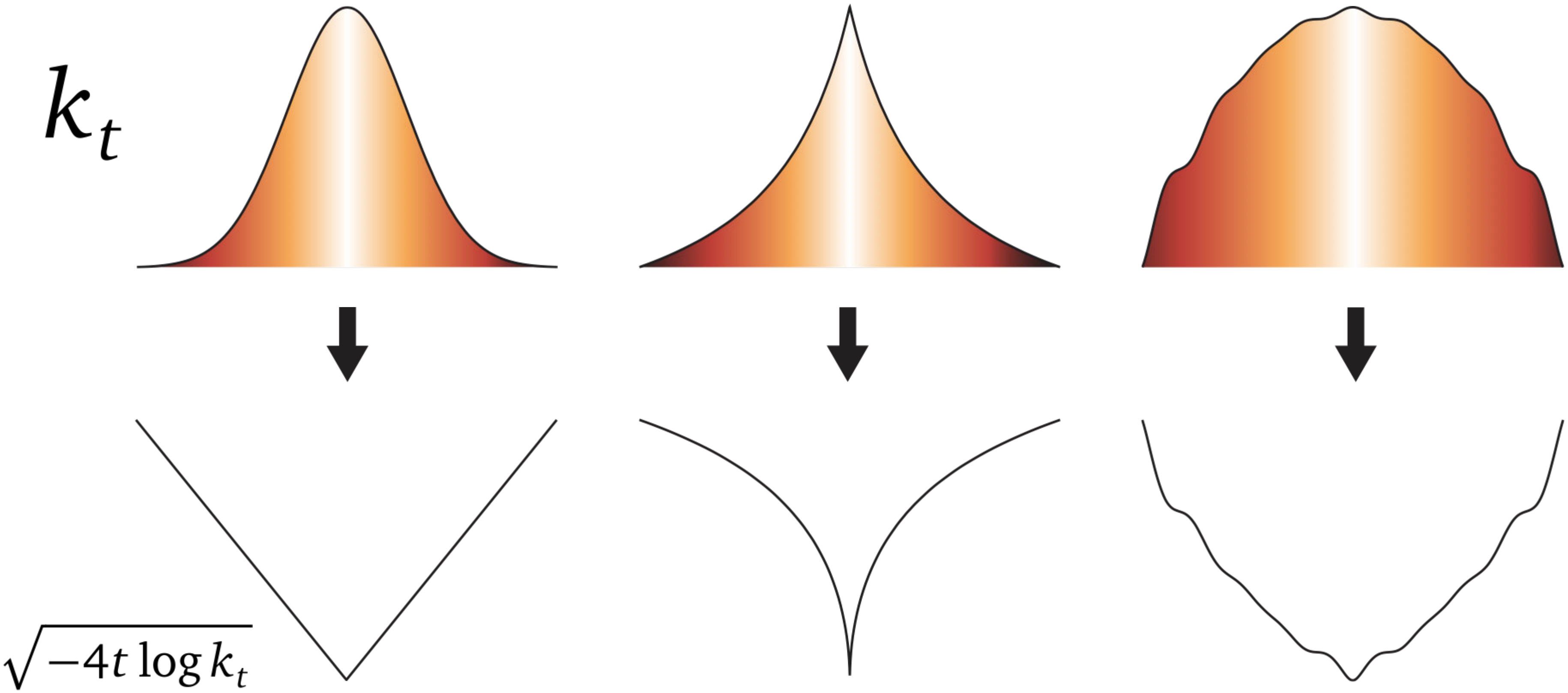
$$\phi = \lim_{t \to 0} \sqrt{-4t \log k_t}$$

heat kernel



[Varadhan 1967]

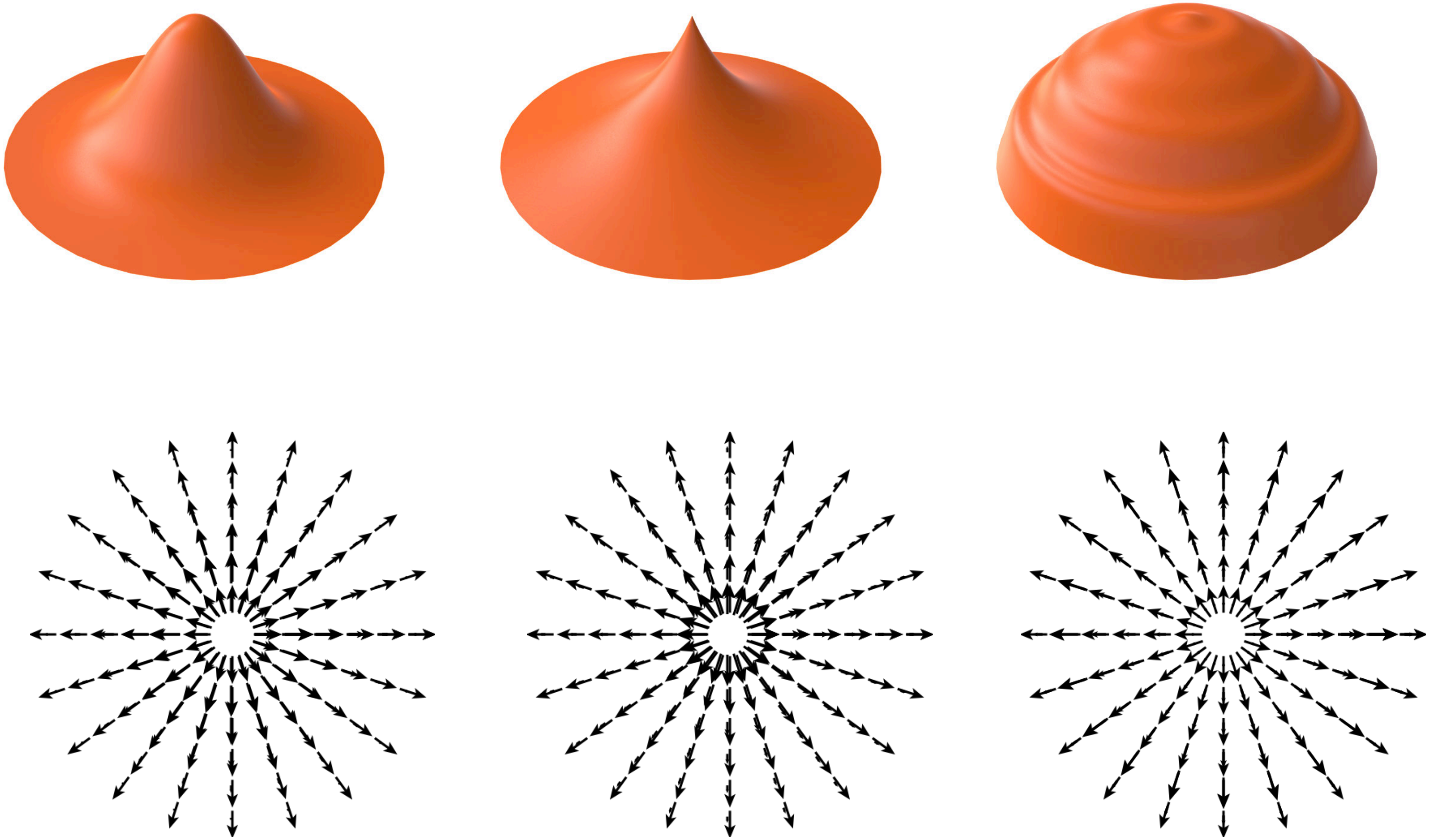
Just apply Varadhan's formula?



Normalizing the gradient

u

$$\frac{-\nabla u}{|\nabla u|}$$



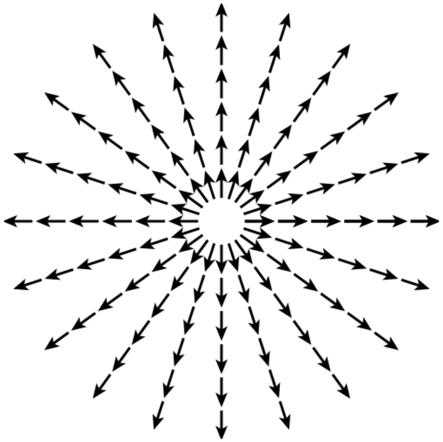
Eikonal: $|\nabla \phi| = 1$

Recovering distance

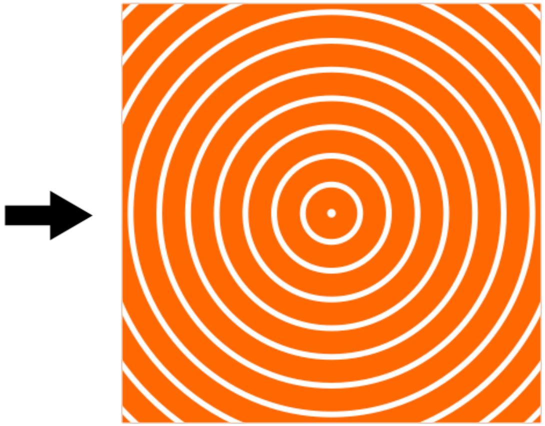
$$\min_{\phi} ||\nabla \phi - X||^2$$

$$\Longleftrightarrow$$

$$\Delta \phi = \nabla \cdot X$$

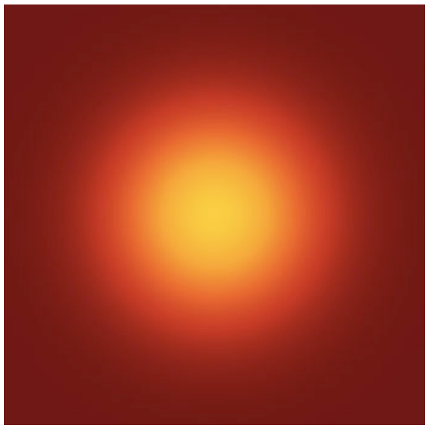
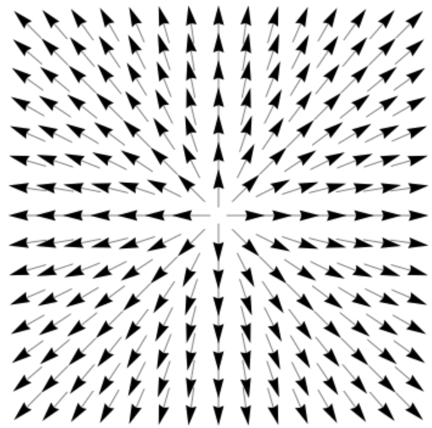


X



ϕ

The Heat Method

Linear	Easy	Linear
		
u	X	ϕ
$\dot{u} = \Delta u$	$X = -\frac{\nabla u}{\lvert\nabla u\rvert}$	$\Delta\phi = \nabla \cdot X$

Temporal Discretization

$$\dot{u} = \Delta u$$

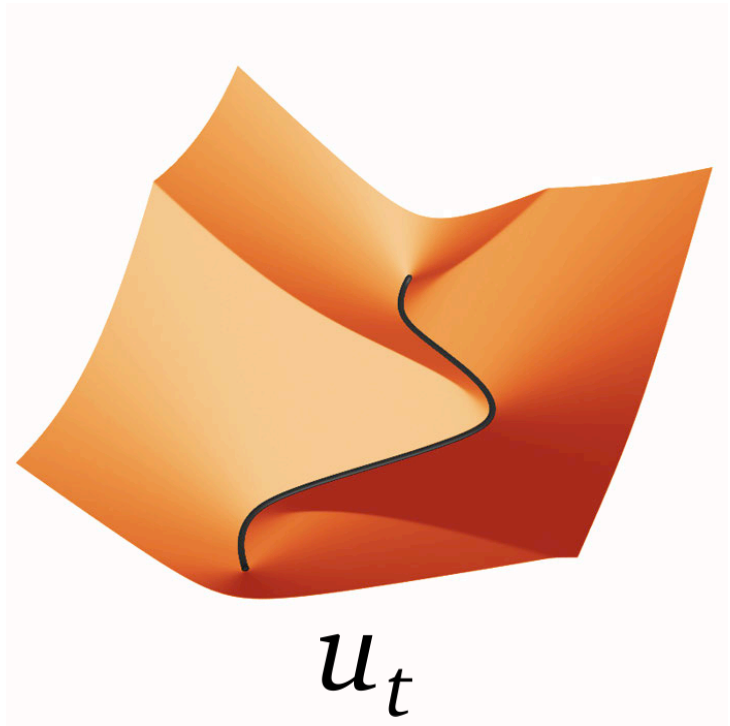
heat equation

$$\frac{u_t - u_0}{t} = \Delta u_t$$

backward Euler

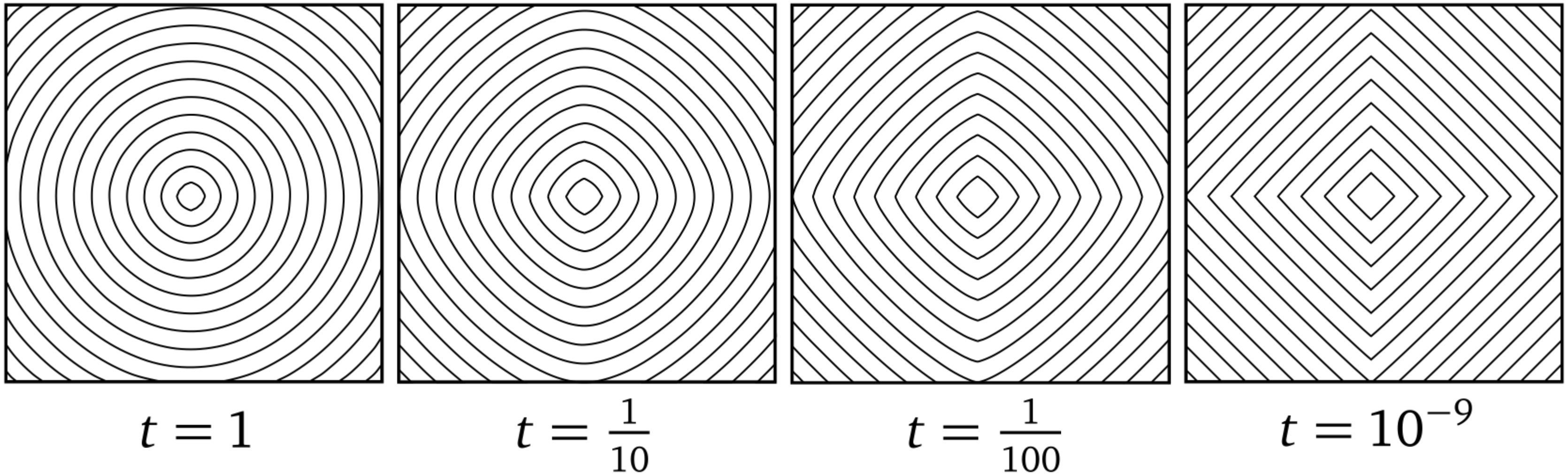
$$(\mathrm{id} - t\Delta)u_t = u_0$$

linear elliptic equation



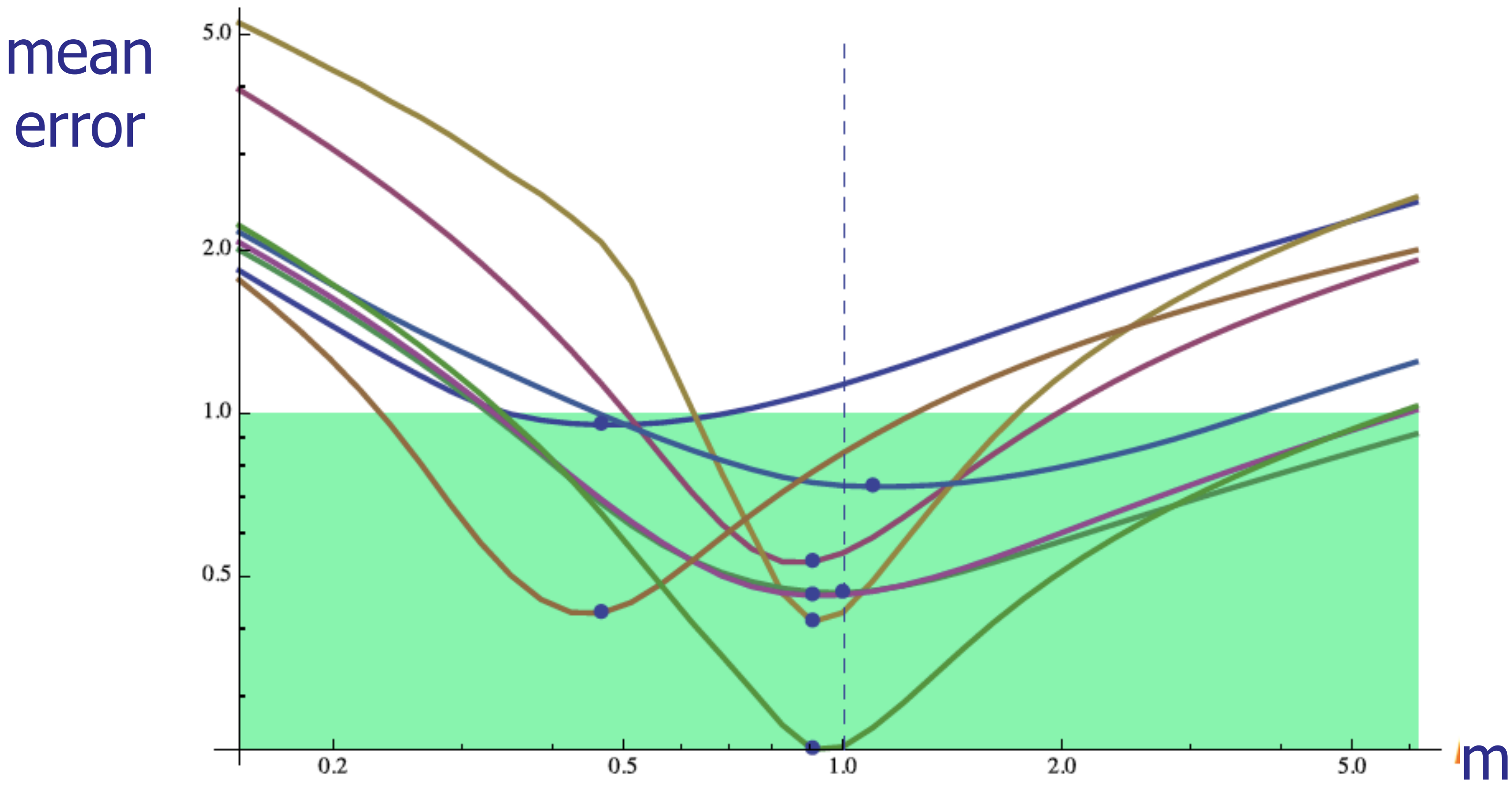
Choosing t

$$\phi = \lim_{t \to 0} -\frac{1}{2}\sqrt{t}\log u_t$$



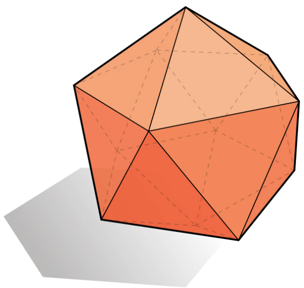
$$t = mh^2$$

Choosing t



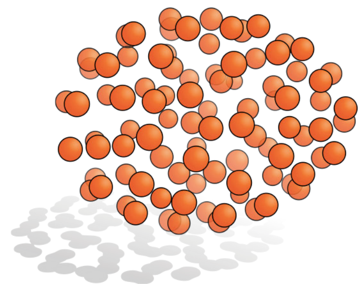
$$t = h^2$$

Spatial Discretization



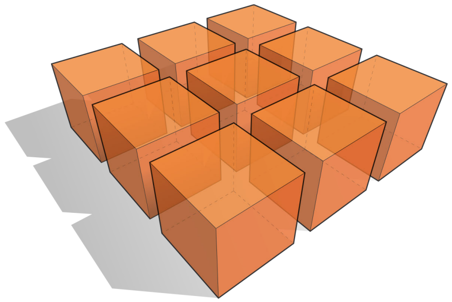
Triangle Meshes

[MacNeal 1949]



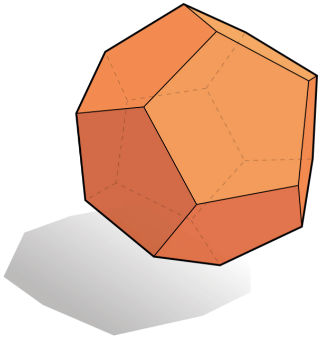
Point Clouds

[Liu et al 2012]



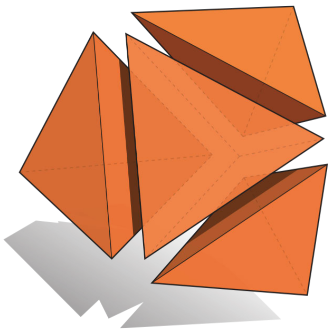
Regular Grids

[Newton 1693]



Polygon Meshes

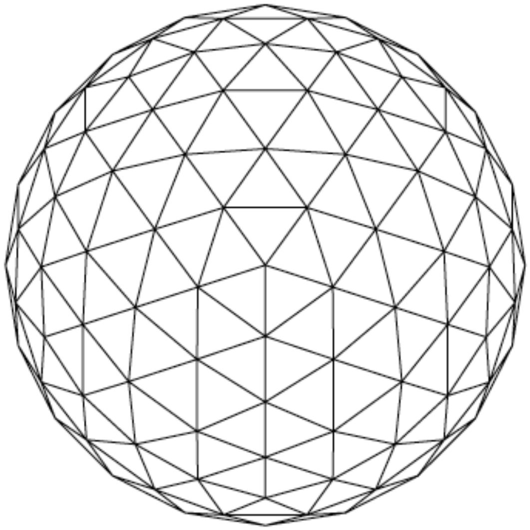
[Alexa & W. 2011]

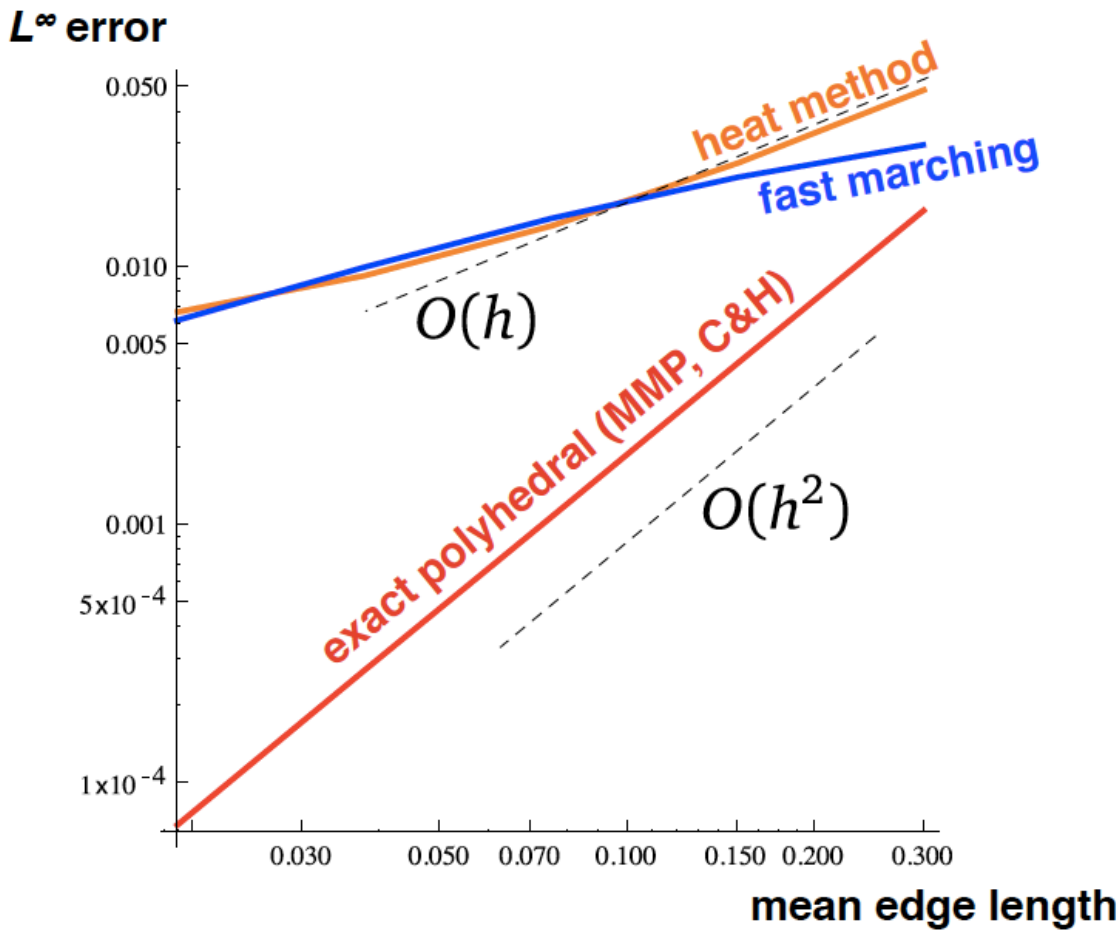


Tet Meshes

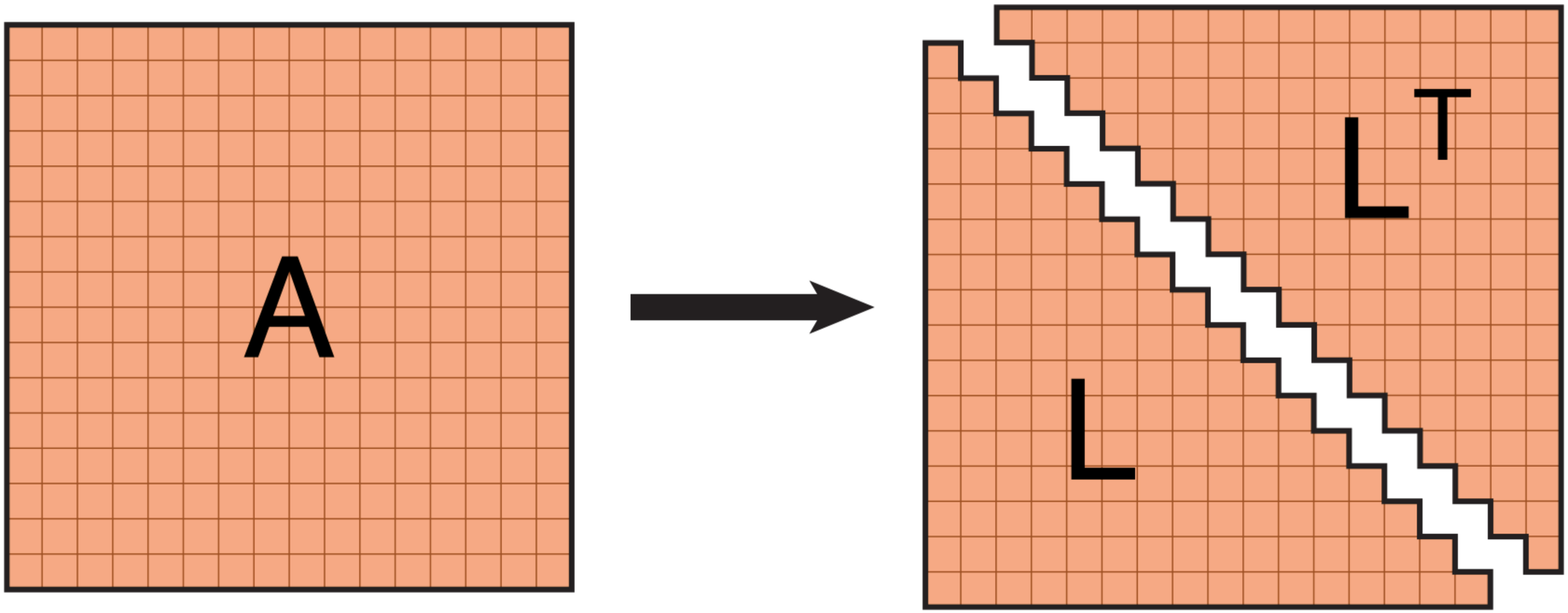
[Desbrun et al 2008]

Convergence

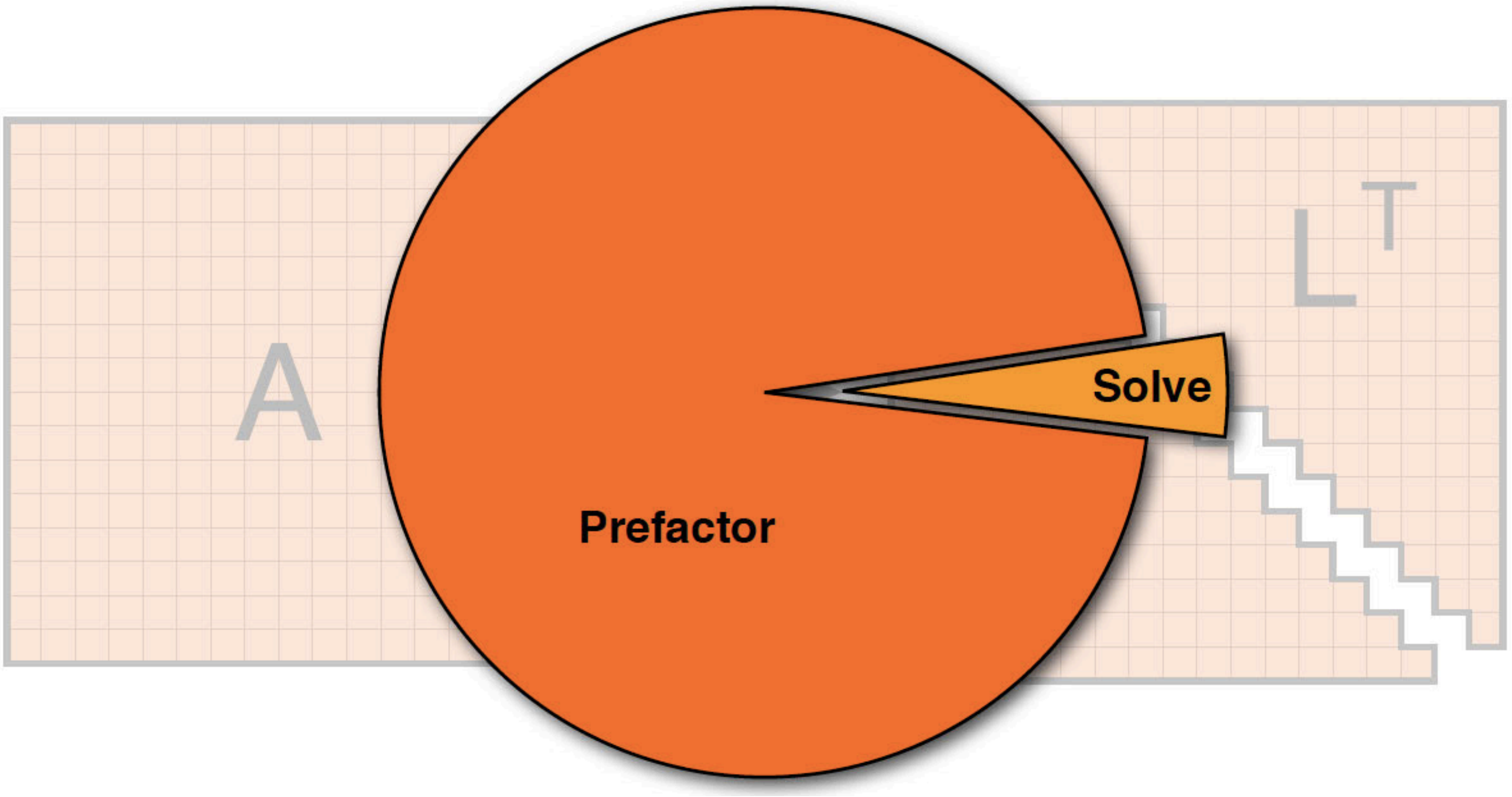




Prefactorization



Prefactorization



Performance

Model	Faces	Heat Method				Fast Marching		
		Factor	Solve	Max	Mean	Time	Max	Mean
Bunny	28k	0.29s	0.02s	1.65%	**0.74%**	**0.28s**	**1.05%**	1.16%
Isis	93k	1.27s	0.11s	1.29%	**0.54%**	**1.08s**	**0.61%**	0.85%
Horse	96k	0.99s	0.07s	1.16%	**0.38%**	**1.01s**	**0.76%**	0.73%
Aphrodite	106k	**1.13s**	**0.08s**	1.95%	**0.93%**	2.38s	**0.90%**	1.04%
Bimba	149k	**2.45s**	**0.15s**	1.35%	0.90%	2.78s	**0.61%**	**0.65%**
Lion	353k	**7.05s**	**0.37s**	**0.68%**	**0.44%**	10.93s	0.74%	0.68%
Ramses	1.6M	**26.47s**	**1.27s**	1.59%	**0.46%**	104.86s	**0.42%**	0.47%

Visual Comparison of Accuracy

Visual Comparison of Accuracy

fast marching

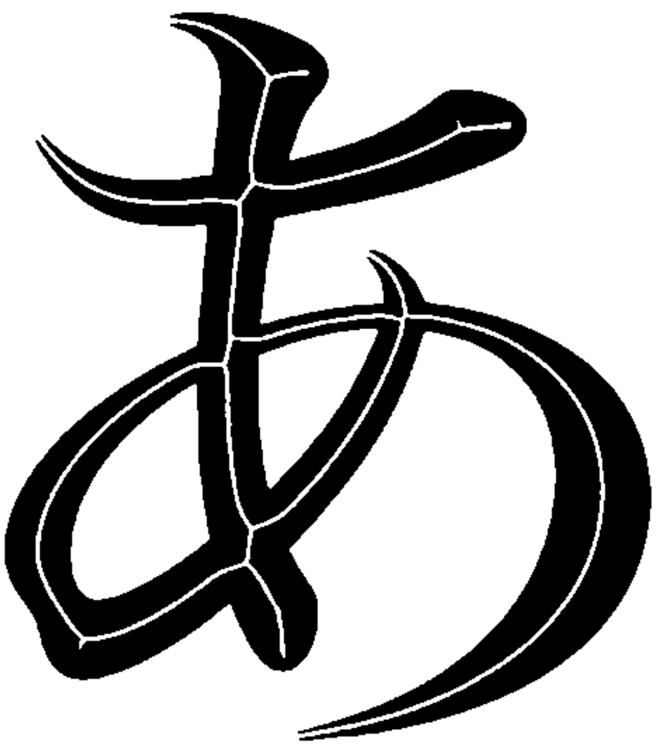
heat method

exact polyhedral

Visual Comparison of Accuracy

fast marching | heat method | exact polyhedral

Medial Axis

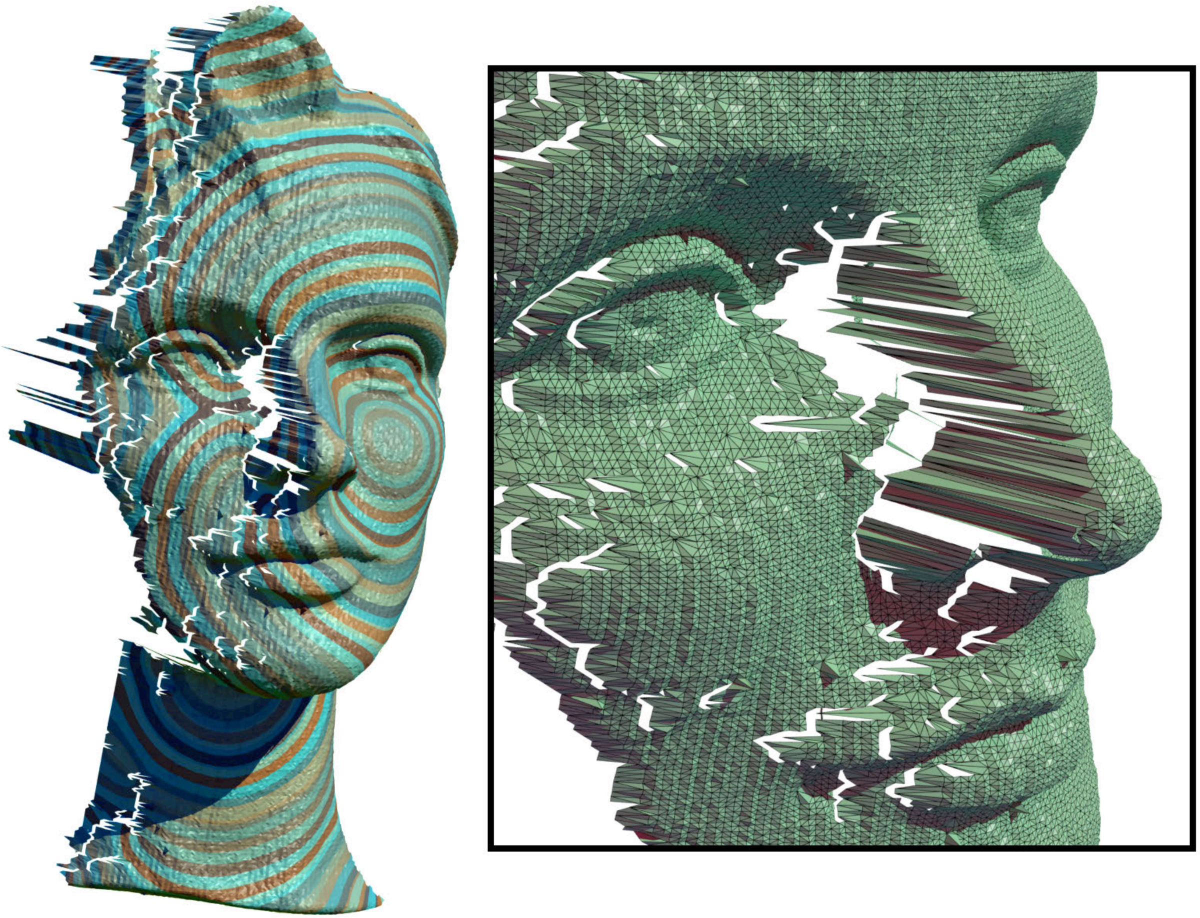


fast marching

heat method

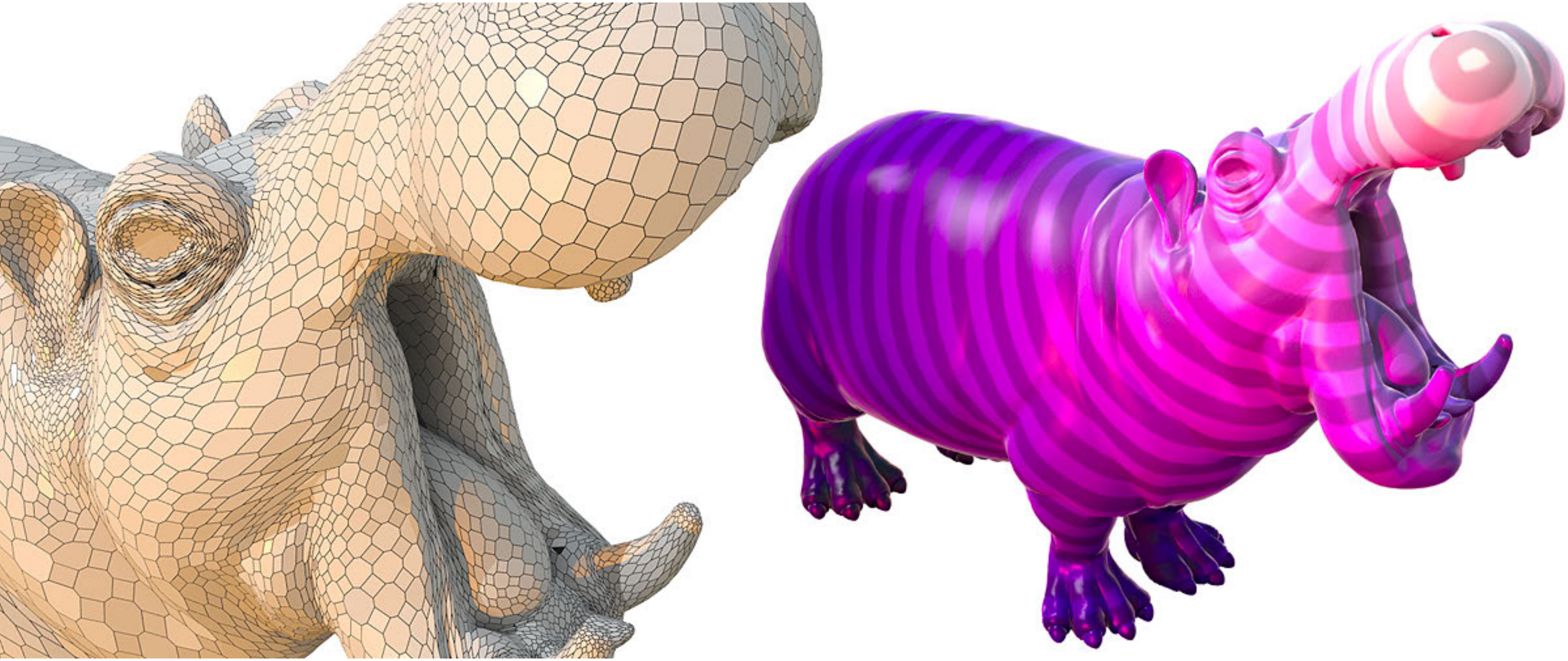
Example: Distance to boundary

Example: Robustness

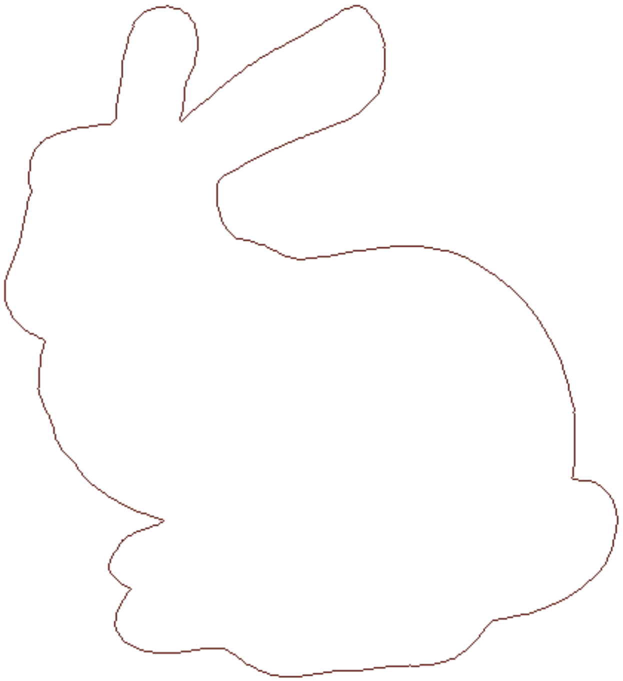


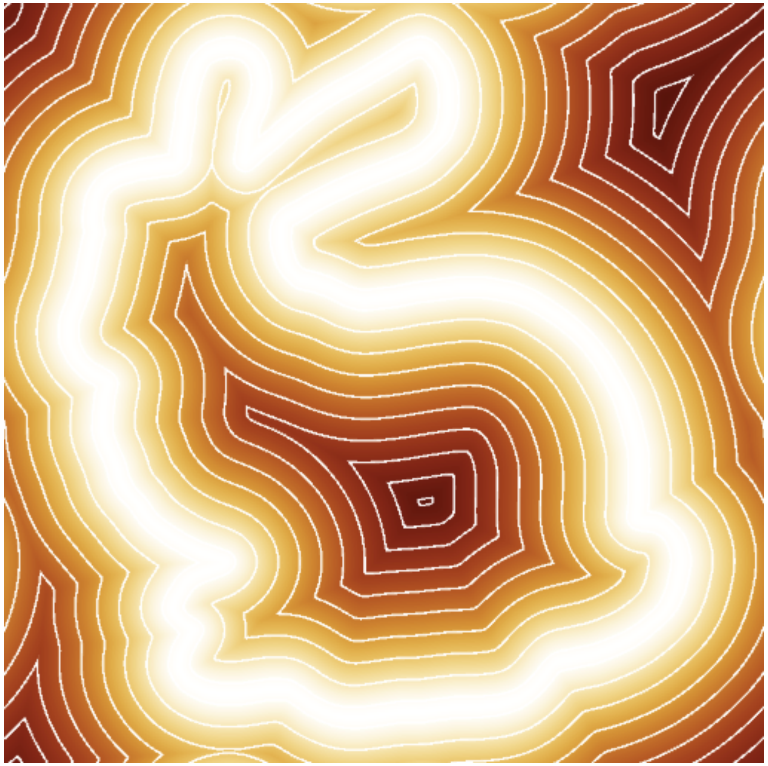
Example: Point Cloud

Example: Polygonal Mesh

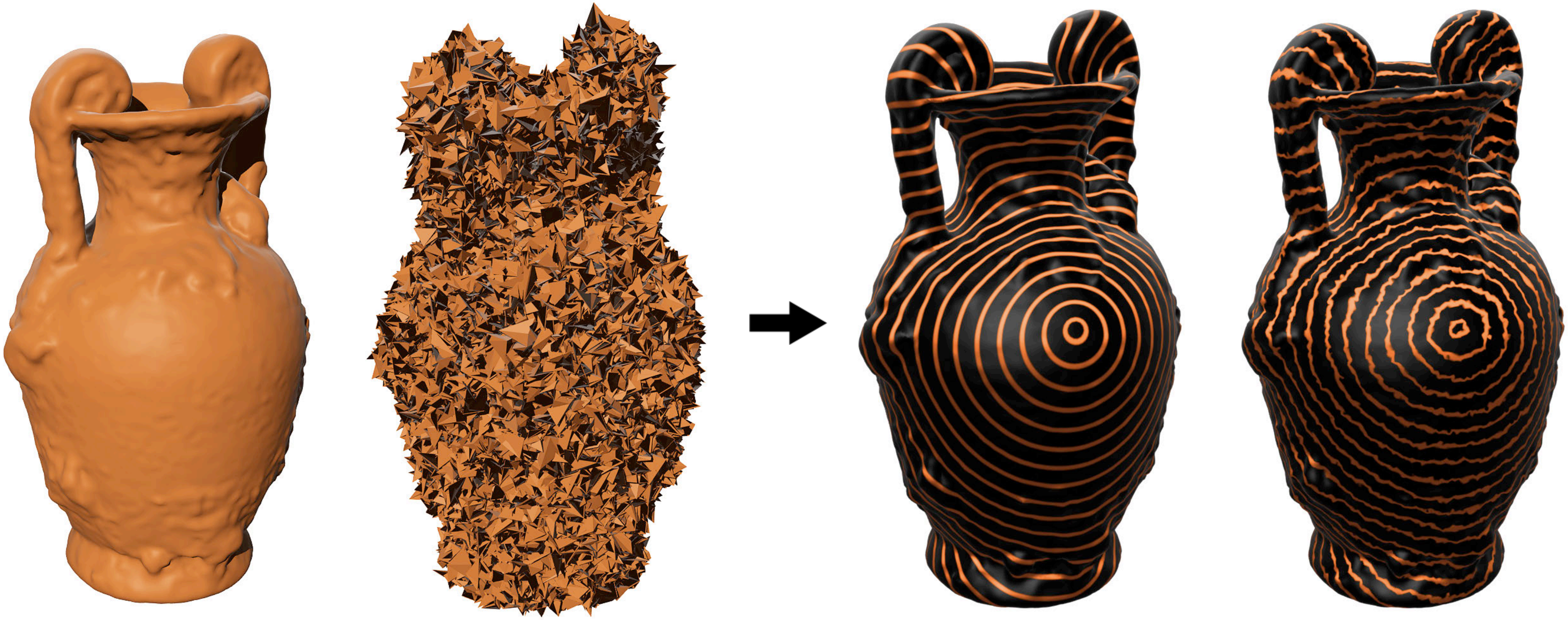


Example: Regular Grid

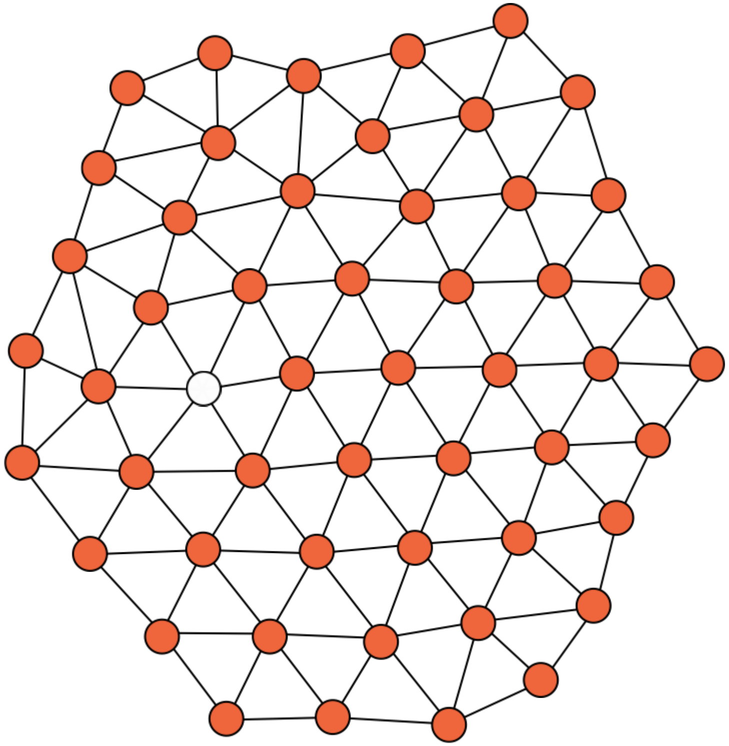


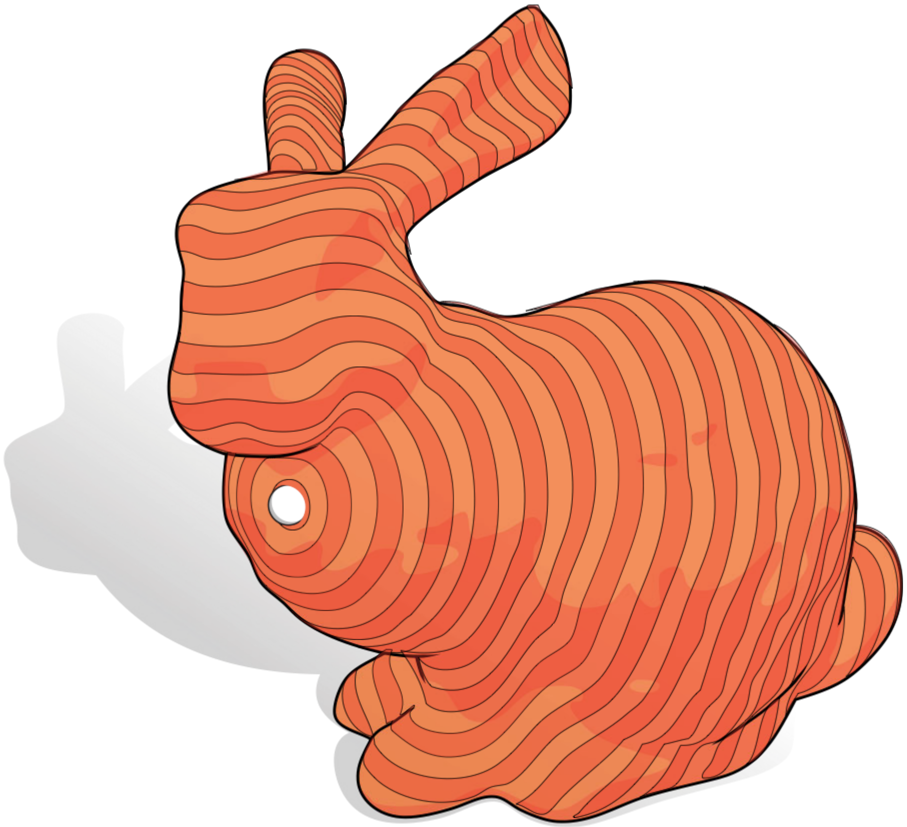


Example: Noise

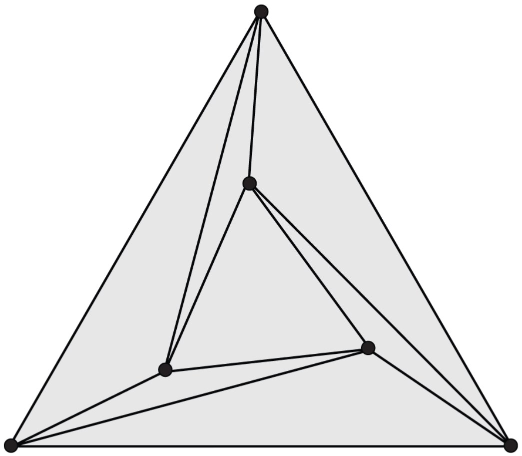


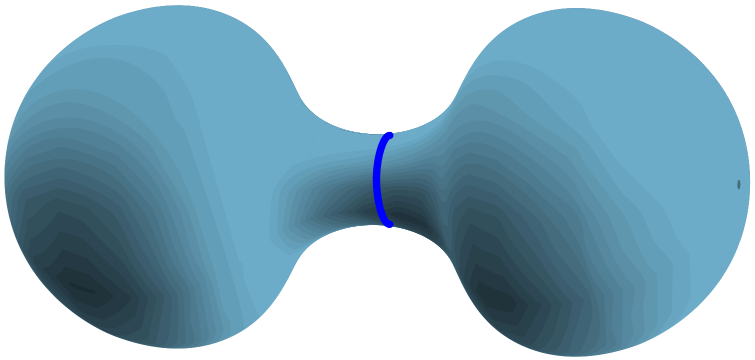
Challenge

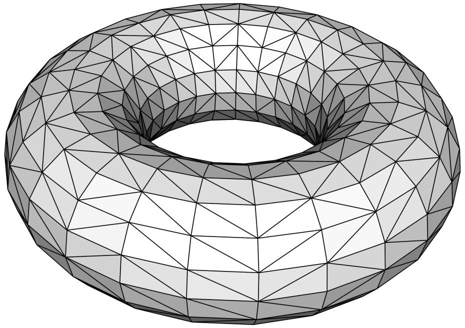


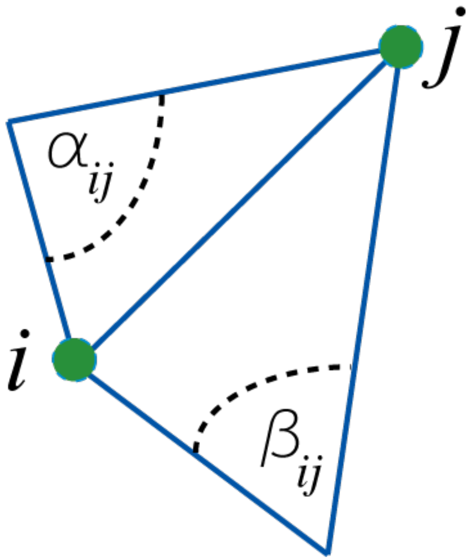


Show convergence of heat method under refinement.









Thank you!