IT575 Computational Shape Modeling
Module - 3: Discrete Geometry

Aditya Tatu

Sampling a Surface

- Let M be a surface. A finite subset $P \in M$ is called a *sample set* of M, and the set P is often referred to as a *point cloud*.

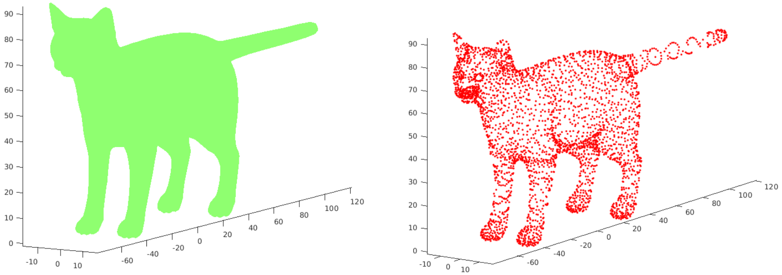


Figure : (left) Original Surface, (right) Point cloud representation.

Quality of Sampling

- Measure of uniformity: A sample-set P is said to be an *r-covering* of the surface M if

$$\cup_{p \in P} B_r(p) = M \quad \Leftrightarrow \quad \forall x \in M, \quad d_M(x, P) \leq r.$$

  - Issue: If P is an r-covering, is it a minimal set?
- A sample set P is said to be *r-separated* if $d_M(p, q) \geq r, \ \forall p, q \in P$, with $p \neq q$.

Quality of Sampling

- Measure of uniformity: A sample-set P is said to be an *r-covering* of the surface M if

$$\cup_{p\in P} B_r(p) = M \quad \Leftrightarrow \quad \forall x \in M, \quad d_M(x,P) \le r.$$

  - Issue: If P is an r-covering, is it a minimal set?

- A sample set P is said to be *r-separated* if $d_M(p,q) \ge r, \ \forall p,q \in P$, with $p \neq q$.

Quality of Sampling

- Measure of uniformity: A sample-set P is said to be an *r-covering* of the surface M if

$$\cup_{p \in P} B_r(p) = M \quad \Leftrightarrow \quad \forall x \in M, \quad d_M(x, P) \leq r.$$

  - Issue: If P is an r-covering, is it a minimal set?
- A sample set P is said to be *r-separated* if $d_M(p, q) \geq r, \; \forall p, q \in P$, with $p \neq q$.

Sampling strategy: Farthest Point Sampling

- Let $p_1 = x_1 \in M$, and $P = \{p_1\}$.
- How should the second point be selected?
- Let $p_2 = \arg\max_{x \in M} d_M(x, P)$, and $P = \{p_1, p_2\}$.
- P is an ______-separated set and an ______ -covering set.
- Similarly, $p_3 = \arg\max_{x \in M} d_M(x, P)$, and $P = P \cup \{p_3\}$.
- Let $P = \{p_1, \ldots, p_n\}$ be the sample set. It is a ______-separated and ______-covering sample set.
- Issues with Farthest point sampling?

Sampling strategy: Farthest Point Sampling

- Let $p_1 = x_1 \in M$, and $P = \{p_1\}$.
- How should the second point be selected?
- Let $p_2 = \arg\max_{x \in M} d_M(x, P)$, and $P = \{p_1, p_2\}$.
- P is an _____-separated set and an _____ -covering set.
- Similarly, $p_3 = \arg\max_{x \in M} d_M(x, P)$, and $P = P \cup \{p_3\}$.
- Let $P = \{p_1, \ldots, p_n\}$ be the sample set. It is a _____-separated and _____-covering sample set.

- Issues with Farthest point sampling?

Sampling strategy: Farthest Point Sampling

- Let $p_1 = x_1 \in M$, and $P = \{p_1\}$.
- How should the second point be selected?
- Let $p_2 = \arg\max_{x \in M} d_M(x, P)$, and $P = \{p_1, p_2\}$.
- P is an ______-separated set and an ______ -covering set.
- Similarly, $p_3 = \arg\max_{x \in M} d_M(x, P)$, and $P = P \cup \{p_3\}$.
- Let $P = \{p_1, \ldots, p_n\}$ be the sample set. It is a ______-separated and ______-covering sample set.

- Issues with Farthest point sampling?

Sampling strategy: Farthest Point Sampling

- Let $p_1 = x_1 \in M$, and $P = \{p_1\}$.
- How should the second point be selected?
- Let $p_2 = \arg\max_{x \in M} d_M(x, P)$, and $P = \{p_1, p_2\}$.
- P is an _____-separated set and an _____ -covering set.
- Similarly, $p_3 = \arg\max_{x \in M} d_M(x, P)$, and $P = P \cup \{p_3\}$.
- Let $P = \{p_1, \ldots, p_n\}$ be the sample set. It is a _____-separated and _____-covering sample set.

- Issues with Farthest point sampling?

Sampling strategy: Farthest Point Sampling

- Let $p_1 = x_1 \in M$, and $P = \{p_1\}$.
- How should the second point be selected?
- Let $p_2 = \arg\max_{x \in M} d_M(x, P)$, and $P = \{p_1, p_2\}$.
- P is an _____-separated set and an _____ -covering set.
- Similarly, $p_3 = \arg\max_{x \in M} d_M(x, P)$, and $P = P \cup \{p_3\}$.
- Let $P = \{p_1, \ldots, p_n\}$ be the sample set. It is a _____-separated and _____-covering sample set.

- Issues with Farthest point sampling?

Sampling strategy: Farthest Point Sampling

- Let $p_1 = x_1 \in M$, and $P = \{p_1\}$.
- How should the second point be selected?
- Let $p_2 = \arg\max_{x \in M} d_M(x, P)$, and $P = \{p_1, p_2\}$.
- P is an _____-separated set and an _____ -covering set.
- Similarly, $p_3 = \arg\max_{x \in M} d_M(x, P)$, and $P = P \cup \{p_3\}$.
- Let $P = \{p_1, \ldots, p_n\}$ be the sample set. It is a _____-separated and _____-covering sample set.
- Issues with Farthest point sampling?

Sampling strategy: Farthest Point Sampling

- Let $p_1 = x_1 \in M$, and $P = \{p_1\}$.
- How should the second point be selected?
- Let $p_2 = \arg\max_{x \in M} d_M(x, P)$, and $P = \{p_1, p_2\}$.
- P is an _____-separated set and an _____ -covering set.
- Similarly, $p_3 = \arg\max_{x \in M} d_M(x, P)$, and $P = P \cup \{p_3\}$.
- Let $P = \{p_1, \ldots, p_n\}$ be the sample set. It is a _____-separated and _____-covering sample set.
- Issues with Farthest point sampling?

Voronoi Tessellation

- Let P be a sample-set of the surface M. Every sample point represents a small nbhd of M.
- For $p_i \in P$, let $V_i(P) = \{x \in M \mid d_M(x, p_i) < d_M(x, p_j), j \neq i, p_j \in P\}$.
- $V_i(P)$ is called the *Voronoi region* of p_i with respect to the sample-set P, and the collection of Voronoi regions of M is called the *Voronoi decomposition* of M generated by P.
- If for $x \in M$, $d_M(x, p_i) = d_M(x, p_j)$, x is said to be on the *Voronoi edge* V_{ij} of the Voronoi regions V_i, V_j.
- If for $x \in M$, $d_M(x, p_i) = d_M(x, p_j) = d_M(x, p_k)$, x is said to be a *Voronoi vertex* V_{ijk} of the Voronoi regions V_i, V_j, V_k.
- Demo: `http://alexbeutel.com/webgl/voronoi.html`.

Voronoi Tessellation

- Let P be a sample-set of the surface M. Every sample point represents a small nbhd of M.
- For $p_i \in P$, let $V_i(P) = \{x \in M \mid d_M(x, p_i) < d_M(x, p_j), j \neq i, p_j \in P\}$.
- $V_i(P)$ is called the *Voronoi region* of p_i with respect to the sample-set P, and the collection of Voronoi regions of M is called the *Voronoi decomposition* of M generated by P.
- If for $x \in M$, $d_M(x, p_i) = d_M(x, p_j)$, x is said to be on the *Voronoi edge* V_{ij} of the Voronoi regions V_i, V_j.
- If for $x \in M$, $d_M(x, p_i) = d_M(x, p_j) = d_M(x, p_k)$, x is said to be a *Voronoi vertex* V_{ijk} of the Voronoi regions V_i, V_j, V_k.
- Demo: `http://alexbeutel.com/webgl/voronoi.html`.

Voronoi Tessellation

- Let P be a sample-set of the surface M. Every sample point represents a small nbhd of M.
- For $p_i \in P$, let
  $V_i(P) = \{x \in M \mid d_M(x, p_i) < d_M(x, p_j), j \neq i, p_j \in P\}$.
- $V_i(P)$ is called the *Voronoi region* of p_i with respect to the sample-set P, and the collection of Voronoi regions of M is called the *Voronoi decomposition* of M generated by P.
- If for $x \in M$, $d_M(x, p_i) = d_M(x, p_j)$, x is said to be on the *Voronoi edge* V_{ij} of the Voronoi regions V_i, V_j.
- If for $x \in M$, $d_M(x, p_i) = d_M(x, p_j) = d_M(x, p_k)$, x is said to be a *Voronoi vertex* V_{ijk} of the Voronoi regions V_i, V_j, V_k.
- Demo: `http://alexbeutel.com/webgl/voronoi.html`.

Voronoi Tessellation

- Let P be a sample-set of the surface M. Every sample point represents a small nbhd of M.
- For $p_i \in P$, let $V_i(P) = \{x \in M \mid d_M(x, p_i) < d_M(x, p_j), j \neq i, p_j \in P\}$.
- $V_i(P)$ is called the *Voronoi region* of p_i with respect to the sample-set P, and the collection of Voronoi regions of M is called the *Voronoi decomposition* of M generated by P.
- If for $x \in M, d_M(x, p_i) = d_M(x, p_j)$, x is said to be on the *Voronoi edge* V_{ij} of the Voronoi regions V_i, V_j.
- If for $x \in M, d_M(x, p_i) = d_M(x, p_j) = d_M(x, p_k)$, x is said to be a *Voronoi vertex* V_{ijk} of the Voronoi regions V_i, V_j, V_k.
- Demo: `http://alexbeutel.com/webgl/voronoi.html`.

Voronoi Tessellation

- Let P be a sample-set of the surface M. Every sample point represents a small nbhd of M.
- For $p_i \in P$, let
  $V_i(P) = \{x \in M \mid d_M(x, p_i) < d_M(x, p_j), j \neq i, p_j \in P\}$.
- $V_i(P)$ is called the *Voronoi region* of p_i with respect to the sample-set P, and the collection of Voronoi regions of M is called the *Voronoi decomposition* of M generated by P.
- If for $x \in M, d_M(x, p_i) = d_M(x, p_j)$, x is said to be on the *Voronoi edge* V_{ij} of the Voronoi regions V_i, V_j.
- If for $x \in M, d_M(x, p_i) = d_M(x, p_j) = d_M(x, p_k)$, x is said to be a *Voronoi vertex* V_{ijk} of the Voronoi regions V_i, V_j, V_k.
- Demo: `http://alexbeutel.com/webgl/voronoi.html`.

Voronoi Tessellation

- Let P be a sample-set of the surface M. Every sample point represents a small nbhd of M.
- For $p_i \in P$, let
  $V_i(P) = \{x \in M \mid d_M(x, p_i) < d_M(x, p_j), j \neq i, p_j \in P\}$.
- $V_i(P)$ is called the *Voronoi region* of p_i with respect to the sample-set P, and the collection of Voronoi regions of M is called the *Voronoi decomposition* of M generated by P.
- If for $x \in M, d_M(x, p_i) = d_M(x, p_j)$, x is said to be on the *Voronoi edge* V_{ij} of the Voronoi regions V_i, V_j.
- If for $x \in M, d_M(x, p_i) = d_M(x, p_j) = d_M(x, p_k)$, x is said to be a *Voronoi vertex* V_{ijk} of the Voronoi regions V_i, V_j, V_k.
- Demo: `http://alexbeutel.com/webgl/voronoi.html`.

- Let $\bar{V}_i$ denote the closure of V_i. Note $M = \cup_i \bar{V}_i$, and $V_i \cap V_j = \emptyset$.
- The decomposition of M into $\{V_i \mid p_i \in P\}$ such that $M = \cup_i V_i$, and $V_i \cap V_j = \emptyset$ is called a *tessellation or cell complex* of M.
  - If the tiles V_i are Voronoi regions, we call it a *Voronoi tessellation*.

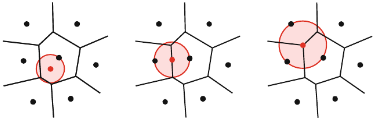


Figure : Voronoi cells. Image from: *Bronstein, Bronstein, Kimmel, Numerical Geometry of Nonrigid shapes, Springer 2008*, henceforth - BBK08

- Let $\bar{V}_i$ denote the closure of V_i. Note $M = \cup_i \bar{V}_i$, and $V_i \cap V_j = \emptyset$.
- The decomposition of M into $\{V_i \mid p_i \in P\}$ such that $M = \cup_i V_i$, and $V_i \cap V_j = \emptyset$ is called a *tessellation or cell complex* of M.
  - If the tiles V_i are Voronoi regions, we call it a *Voronoi tessellation*.

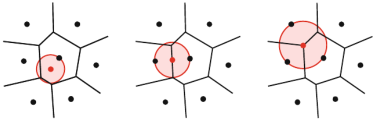


Figure : Voronoi cells. Image from: *Bronstein, Bronstein, Kimmel, Numerical Geometry of Nonrigid shapes, Springer 2008*, henceforth - BBK08

- Let $\bar{V}_i$ denote the closure of V_i. Note $M = \cup_i \bar{V}_i$, and $V_i \cap V_j = \emptyset$.
- The decomposition of M into $\{V_i \mid p_i \in P\}$ such that $M = \cup_i V_i$, and $V_i \cap V_j = \emptyset$ is called a *tessellation or cell complex* of M.
- If the tiles V_i are Voronoi regions, we call it a *Voronoi tessellation*.

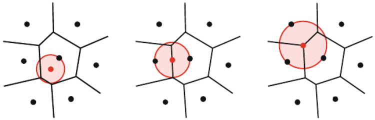


Figure : Voronoi cells. Image from: *Bronstein, Bronstein, Kimmel, Numerical Geometry of Nonrigid shapes, Springer 2008*, henceforth - BBK08

.

Sampling and Voronoi tessellation

- Let P be a sample set of a surface M.
- Let $s : M \to P$ be defined by

$$s(x) = p_i \quad \text{where } p_i \text{ is such that } x \in V_i(M).$$

- By representing M by P, $\forall x \in M$, error incorporated is $e(x) = d_M(x, s(x) = p_i)$.
- Total error: $e(P) = \int_M d_M^2(x, s(x)) = \sum_{i=1}^{N} \int_{V_i(P)} d_M^2(x, p_i)\ da$
  - Thus, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} d_M^2(x, x^*)\ da$.
  - If $M \subset \mathbb{R}^2$, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} ||x - x^*||^2\ dx$.
  - Thus, $p_i = \frac{\int_{V_i} x\ dx}{\int_{V_i} dx}$ is the *centroid* of the Voronoi cell V_i.

Sampling and Voronoi tessellation

- Let P be a sample set of a surface M.
- Let $s : M \to P$ be defined by

$$s(x) = p_i \ \text{ where } p_i \text{ is such that } x \in V_i(M).$$

- By representing M by P, $\forall x \in M$, error incorporated is $e(x) = d_M(x, s(x) = p_i)$.
- Total error:
  $e(P) = \int_M d_M^2(x, s(x)) = \sum_{i=1}^{N} \int_{V_i(P)} d_M^2(x, p_i) \ da$
  - Thus, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} d_M^2(x, x^*) \ da$.
  - If $M \subset \mathbb{R}^2$, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} ||x - x^*||^2 \ dx$.
  - Thus, $p_i = \frac{\int_{V_i} x \ dx}{\int_{V_i} dx}$ is the *centroid* of the Voronoi cell V_i.

Sampling and Voronoi tessellation

- Let P be a sample set of a surface M.
- Let $s : M \to P$ be defined by

$$s(x) = p_i \text{ where } p_i \text{ is such that } x \in V_i(M).$$

- By representing M by P, $\forall x \in M$, error incorporated is $e(x) = d_M(x, s(x) = p_i)$.
- Total error: $e(P) = \int_M d_M^2(x, s(x)) = \sum_{i=1}^{N} \int_{V_i(P)} d_M^2(x, p_i)\ da$
  - Thus, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} d_M^2(x, x^*)\ da$.
  - If $M \subset \mathbb{R}^2$, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} ||x - x^*||^2\ dx$.
  - Thus, $p_i = \frac{\int_{V_i} x\ dx}{\int_{V_i} dx}$ is the *centroid* of the Voronoi cell V_i.

Sampling and Voronoi tessellation

- Let P be a sample set of a surface M.
- Let $s : M \to P$ be defined by

$$s(x) = p_i \;\text{ where } p_i \text{ is such that } x \in V_i(M).$$

- By representing M by P, $\forall x \in M$, error incorporated is $e(x) = d_M(x, s(x) = p_i)$.
- Total error: $e(P) = \int_M d_M^2(x, s(x)) = \sum_{i=1}^{N} \int_{V_i(P)} d_M^2(x, p_i) \; da$
- Thus, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} d_M^2(x, x^*) \; da$.
- If $M \subset \mathbb{R}^2$, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} ||x - x^*||^2 \; dx$.
- Thus, $p_i = \frac{\int_{V_i} x \; dx}{\int_{V_i} dx}$ is the *centroid* of the Voronoi cell V_i.

Sampling and Voronoi tessellation

- Let P be a sample set of a surface M.
- Let $s : M \to P$ be defined by

$$s(x) = p_i \text{ where } p_i \text{ is such that } x \in V_i(M).$$

- By representing M by P, $\forall x \in M$, error incorporated is $e(x) = d_M(x, s(x) = p_i)$.
- Total error: $e(P) = \int_M d_M^2(x, s(x)) = \sum_{i=1}^N \int_{V_i(P)} d_M^2(x, p_i)\ da$
- Thus, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} d_M^2(x, x^*)\ da$.
- If $M \subset \mathbb{R}^2$, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} ||x - x^*||^2\ dx$.
- Thus, $p_i = \frac{\int_{V_i} x\ dx}{\int_{V_i} dx}$ is the *centroid* of the Voronoi cell V_i.

Sampling and Voronoi tessellation

- Let P be a sample set of a surface M.
- Let $s : M \to P$ be defined by

$$s(x) = p_i \ \text{ where } p_i \text{ is such that } x \in V_i(M).$$

- By representing M by P, $\forall x \in M$, error incorporated is $e(x) = d_M(x, s(x) = p_i)$.
- Total error:
  $e(P) = \int_M d_M^2(x, s(x)) = \sum_{i=1}^{N} \int_{V_i(P)} d_M^2(x, p_i) \ da$
   - Thus, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} d_M^2(x, x^*) \ da$.
   - If $M \subset \mathbb{R}^2, p_i = \arg\min_{x^* \in V_i} \int_{V_i} ||x - x^*||^2 \ dx$.
   - Thus, $p_i = \frac{\int_{V_i} x \ dx}{\int_{V_i} \ dx}$ is the *centroid* of the Voronoi cell V_i.

Sampling and Voronoi tessellation

- Let P be a sample set of a surface M.
- Let $s : M \to P$ be defined by

$$s(x) = p_i \text{ where } p_i \text{ is such that } x \in V_i(M).$$

- By representing M by P, $\forall x \in M$, error incorporated is $e(x) = d_M(x, s(x) = p_i)$.
- Total error:
  $e(P) = \int_M d_M^2(x, s(x)) = \sum_{i=1}^N \int_{V_i(P)} d_M^2(x, p_i)\ da$
  - Thus, $p_i = \arg\min_{x^* \in V_i} \int_{V_i} d_M^2(x, x^*)\ da$.
  - If $M \subset \mathbb{R}^2, p_i = \arg\min_{x^* \in V_i} \int_{V_i} ||x - x^*||^2\ dx$.
  - Thus, $p_i = \frac{\int_{V_i} x\ dx}{\int_{V_i}\ dx}$ is the *centroid* of the Voronoi cell V_i.

Lloyd-Max algorithm

- Given a surface M, and an initial sample-set P.
- Construct voronoi tessellation of M generated by P.
- Update: $P =$ centroids of voronoi cells. Repeat till convergence.

Lloyd-Max algorithm

- Given a surface M, and an initial sample-set P.
- Construct voronoi tessellation of M generated by P.
- Update: $P =$ centroids of voronoi cells. Repeat till convergence.

Lloyd-Max algorithm

- Given a surface M, and an initial sample-set P.
- Construct voronoi tessellation of M generated by P.
- Update: $P =$ centroids of voronoi cells. Repeat till convergence.

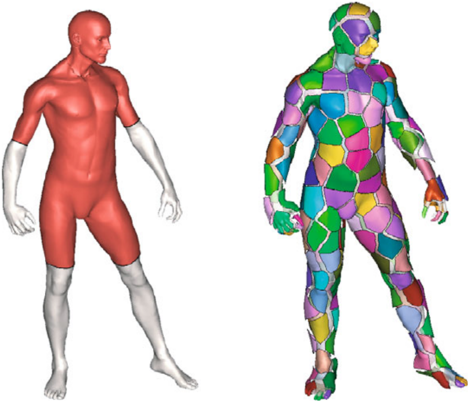


Figure : (left) Inadequate (right) Adequate sampling density. The voronoi regions can be deformed to unit disks in the example on the right. Image from BBK08.

Edge structure

- The sample-set P is a zero-dimensional approximation of the surface M.
- Connect two distinct sample points p_i and p_j if $d_M(p_i, p_j) < r$.
  - The set $N(p_k) = \{p_q \in P \mid d_M(p_k, p_q) < r\}$ is the *Neighborhood* of p_k.
  - Thus P with the above edges gives an *undirected binary graph*.
  - Approximate Edge lengths $l(p_i, p_j) \simeq ||p_i - p_j||_2$.
  - The edges give a one-dimensional approximation of the surface M.

Edge structure

- The sample-set P is a zero-dimensional approximation of the surface M.
- Connect two distinct sample points p_i and p_j if $d_M(p_i, p_j) < r$.
- The set $N(p_k) = \{p_q \in P \mid d_M(p_k, p_q) < r\}$ is the *Neighborhood* of p_k.
- Thus P with the above edges gives an *undirected binary graph*.
- Approximate Edge lengths $l(p_i, p_j) \simeq ||p_i - p_j||_2$.
- The edges give a one-dimensional approximation of the surface M.

Edge structure

- The sample-set P is a zero-dimensional approximation of the surface M.
- Connect two distinct sample points p_i and p_j if $d_M(p_i, p_j) < r$.
- The set $N(p_k) = \{p_q \in P \mid d_M(p_k, p_q) < r\}$ is the *Neighborhood* of p_k.
- Thus P with the above edges gives an *undirected binary graph*.
- Approximate Edge lengths $l(p_i, p_j) \simeq ||p_i - p_j||_2$.
- The edges give a one-dimensional approximation of the surface M.

Edge structure

- The sample-set P is a zero-dimensional approximation of the surface M.
- Connect two distinct sample points p_i and p_j if $d_M(p_i, p_j) < r$.
- The set $N(p_k) = \{p_q \in P \mid d_M(p_k, p_q) < r\}$ is the *Neighborhood* of p_k.
- Thus P with the above edges gives an *undirected binary graph*.
- Approximate Edge lengths $l(p_i, p_j) \simeq ||p_i - p_j||_2$.
- The edges give a one-dimensional approximation of the surface M.

Edge structure

- The sample-set P is a zero-dimensional approximation of the surface M.
- Connect two distinct sample points p_i and p_j if $d_M(p_i, p_j) < r$.
  - The set $N(p_k) = \{p_q \in P \mid d_M(p_k, p_q) < r\}$ is the *Neighborhood* of p_k.
  - Thus P with the above edges gives an *undirected binary graph*.
  - Approximate Edge lengths $l(p_i, p_j) \simeq ||p_i - p_j||_2$.
  - The edges give a one-dimensional approximation of the surface M.

Edge structure

- The sample-set P is a zero-dimensional approximation of the surface M.
- Connect two distinct sample points p_i and p_j if $d_M(p_i, p_j) < r$.
- The set $N(p_k) = \{p_q \in P \mid d_M(p_k, p_q) < r\}$ is the *Neighborhood* of p_k.
- Thus P with the above edges gives an *undirected binary graph*.
- Approximate Edge lengths $l(p_i, p_j) \simeq ||p_i - p_j||_2$.
- The edges give a one-dimensional approximation of the surface M.

Two dimensional approximation

- Let p_i, p_j and p_k be three sample-sets whose Voronoi cells are adjacent to each other and thus yield a vertex V_{ijk}.
- Connect p_i with p_j by a curve of shortest length: Γ_{ij}. Similarly curves Γ_{jk} and Γ_{ki} connect points p_j with p_k and p_k with p_i, resp., forming a *geodesic triangle*.
- Degenerate cases: (a) Three *collinear* points, (b) Four *cocircular* points.
- Decomposition of M into these geodesic triangles is called *Delaunay tessellation*.

Two dimensional approximation

- Let p_i, p_j and p_k be three sample-sets whose Voronoi cells are adjacent to each other and thus yield a vertex V_{ijk}.
- Connect p_i with p_j by a curve of shortest length: Γ_{ij}. Similarly curves Γ_{jk} and Γ_{ki} connect points p_j with p_k and p_k with p_i, resp., forming a *geodesic triangle*.
- Degenerate cases: (a) Three *collinear* points, (b) Four *cocircular* points.
- Decomposition of M into these geodesic triangles is called *Delaunay tessellation*.

Two dimensional approximation

- Let p_i, p_j and p_k be three sample-sets whose Voronoi cells are adjacent to each other and thus yield a vertex V_{ijk}.
- Connect p_i with p_j by a curve of shortest length: Γ_{ij}. Similarly curves Γ_{jk} and Γ_{ki} connect points p_j with p_k and p_k with p_i, resp., forming a *geodesic triangle*.
- Degenerate cases: (a) Three *collinear* points, (b) Four *cocircular* points.
- Decomposition of M into these geodesic triangles is called *Delaunay tessellation*.

Two dimensional approximation

- Let p_i, p_j and p_k be three sample-sets whose Voronoi cells are adjacent to each other and thus yield a vertex V_{ijk}.
- Connect p_i with p_j by a curve of shortest length: Γ_{ij}. Similarly curves Γ_{jk} and Γ_{ki} connect points p_j with p_k and p_k with p_i, resp., forming a *geodesic triangle*.
- Degenerate cases: (a) Three *collinear* points, (b) Four *cocircular* points.
- Decomposition of M into these geodesic triangles is called *Delaunay tessellation*.

Triangular meshes

- We can approximate the geodesic triangles by Euclidean/planar triangles, giving us a triangular mesh:

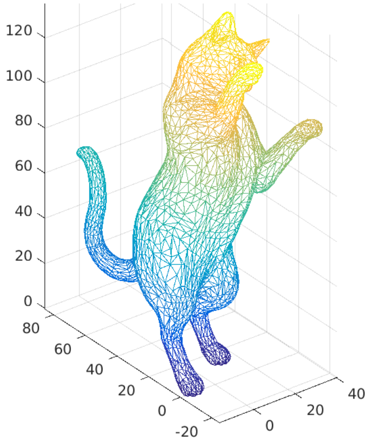


Figure : Triangular mesh surface approximation.

- Triangular meshes are by far the most preferred discretization.
  - Approximate surface locally via planar polygons - we prefer triangles.
  - Optimized software and hardware available for processing triangles.

- Triangular meshes are by far the most preferred discretization.
  - Approximate surface locally via planar polygons - we prefer triangles.
  - Optimized software and hardware available for processing triangles.

- Triangular meshes are by far the most preferred discretization.
  - Approximate surface locally via planar polygons - we prefer triangles.
  - Optimized software and hardware available for processing triangles.

Datastructure for Mesh

- Data: Geometry & Topology.
  - Geometry: For a mesh with n vertices, the geometry consists of an *ordered* $n \times 3$ array of coordinates of the n vertices.
  - Topology: Connectivity information stored as an $m \times 3$ integer array, where m is the number of triangles in the mesh, and each row consisting of indices i, j, k such that the vertices v_i, v_j, v_k form a triangle.
  - A triangle is often referred to as a *face*. The surface represented by the triangular mesh is:

$$M' = \cup_{i=1}^{m} conv(x_{t_i}, y_{t_i}, z_{t_i})$$

  - Additional information may be stored, for example list of edges as a $k \times 2$ array consisting of the index pairs of vertices forming an edge.

Datastructure for Mesh

- Data: Geometry & Topology.
  - Geometry: For a mesh with n vertices, the geometry consists of an *ordered* $n \times 3$ array of coordinates of the n vertices.
  - Topology: Connectivity information stored as an $m \times 3$ integer array, where m is the number of triangles in the mesh, and each row consisting of indices i, j, k such that the vertices v_i, v_j, v_k form a triangle.
  - A triangle is often referred to as a *face*. The surface represented by the triangular mesh is:

$$M' = \cup_{i=1}^{m} conv(x_{t_i}, y_{t_i}, z_{t_i})$$

  - Additional information may be stored, for example list of edges as a $k \times 2$ array consisting of the index pairs of vertices forming an edge.

Datastructure for Mesh

- Data: Geometry & Topology.
  - Geometry: For a mesh with n vertices, the geometry consists of an *ordered* $n \times 3$ array of coordinates of the n vertices.
  - Topology: Connectivity information stored as an $m \times 3$ integer array, where m is the number of triangles in the mesh, and each row consisting of indices i, j, k such that the vertices v_i, v_j, v_k form a triangle.
  - A triangle is often referred to as a *face*. The surface represented by the triangular mesh is:

$$M' = \cup_{i=1}^{m} conv(x_{t_i}, y_{t_i}, z_{t_i})$$

  - Additional information may be stored, for example list of edges as a $k \times 2$ array consisting of the index pairs of vertices forming an edge.

Datastructure for Mesh

- Data: Geometry & Topology.
  - Geometry: For a mesh with n vertices, the geometry consists of an *ordered* $n \times 3$ array of coordinates of the n vertices.
  - Topology: Connectivity information stored as an $m \times 3$ integer array, where m is the number of triangles in the mesh, and each row consisting of indices i, j, k such that the vertices v_i, v_j, v_k form a triangle.
  - A triangle is often referred to as a *face*. The surface represented by the triangular mesh is:

$$M' = \cup_{i=1}^{m} conv(x_{t_i}, y_{t_i}, z_{t_i})$$

  - Additional information may be stored, for example list of edges as a $k \times 2$ array consisting of the index pairs of vertices forming an edge.

Datastructure for Mesh

- Data: Geometry & Topology.
  - Geometry: For a mesh with n vertices, the geometry consists of an *ordered* $n \times 3$ array of coordinates of the n vertices.
  - Topology: Connectivity information stored as an $m \times 3$ integer array, where m is the number of triangles in the mesh, and each row consisting of indices i, j, k such that the vertices v_i, v_j, v_k form a triangle.
  - A triangle is often referred to as a *face*. The surface represented by the triangular mesh is:

$$M' = \cup_{i=1}^{m} conv(x_{t_i}, y_{t_i}, z_{t_i})$$

  - Additional information may be stored, for example list of edges as a $k \times 2$ array consisting of the index pairs of vertices forming an edge.

Degenerate cases

- Non-manifold meshes:

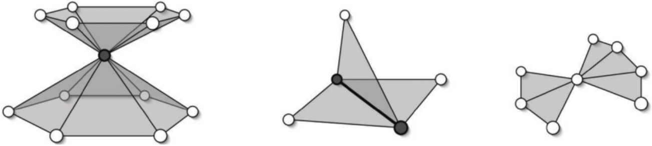


Figure : Examples of non-manifold meshes. Image from *Polygonal Mesh Processing, Botsch et al., A.K.Peters, 2010*, henceforth BKPAL10

- Assumptions: No non-manifold vertices or edges.

Degenerate cases

- Non-manifold meshes:

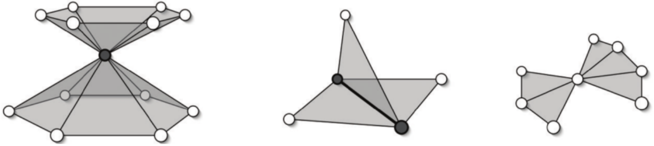


Figure : Examples of non-manifold meshes. Image from *Polygonal Mesh Processing, Botsch et al., A.K.Peters, 2010*, henceforth BKPAL10

- Assumptions: No non-manifold vertices or edges.

Discrete Differential Geometry

Differential geometric properties

- A triangular mesh is a linear approximation of the original surface.
- Most of differential geometry depends on derivatives of different quantities.
- We would want to compute the values of the derivatives of given quantities/signal/function at the vertices.
- Given a manifold M, and a function $f : M \to \mathbb{R}$ on it, the directional derivative of f along $v \in T_pM$ at p is defined as $(D_v f)(p) = \lim_{t \to 0} \frac{f(\gamma(t)) - f(p)}{t}$, where $\gamma : (-\epsilon, \epsilon) \to M$ is a smooth curve on M such that $\gamma(0) = p$ and $\dot{\gamma}(0) = v \in T_pM$.

Differential geometric properties

- A triangular mesh is a linear approximation of the original surface.
- Most of differential geometry depends on derivatives of different quantities.
- We would want to compute the values of the derivatives of given quantities/signal/function at the vertices.
- Given a manifold M, and a function $f : M \to \mathbb{R}$ on it, the directional derivative of f along $v \in T_pM$ at p is defined as $(D_v f)(p) = \lim_{t\to 0} \frac{f(\gamma(t))-f(p)}{t}$, where $\gamma : (-\epsilon, \epsilon) \to M$ is a smooth curve on M such that $\gamma(0) = p$ and $\dot{\gamma}(0) = v \in T_pM$.

Differential geometric properties

- A triangular mesh is a linear approximation of the original surface.
- Most of differential geometry depends on derivatives of different quantities.
- We would want to compute the values of the derivatives of given quantities/signal/function at the vertices.
- Given a manifold M, and a function $f : M \to \mathbb{R}$ on it, the directional derivative of f along $v \in T_pM$ at p is defined as $(D_v f)(p) = \lim_{t \to 0} \frac{f(\gamma(t)) - f(p)}{t}$, where $\gamma : (-\epsilon, \epsilon) \to M$ is a smooth curve on M such that $\gamma(0) = p$ and $\dot{\gamma}(0) = v \in T_pM$.

Differential geometric properties

- A triangular mesh is a linear approximation of the original surface.
- Most of differential geometry depends on derivatives of different quantities.
- We would want to compute the values of the derivatives of given quantities/signal/function at the vertices.
- Given a manifold M, and a function $f : M \to \mathbb{R}$ on it, the directional derivative of f along $v \in T_pM$ at p is defined as $(D_v f)(p) = \lim_{t\to 0} \frac{f(\gamma(t))-f(p)}{t}$, where $\gamma : (-\epsilon, \epsilon) \to M$ is a smooth curve on M such that $\gamma(0) = p$ and $\dot{\gamma}(0) = v \in T_pM$.

Approximating derivatives

- Let us first consider sampling of a 1 D signal as shown in the figure below.

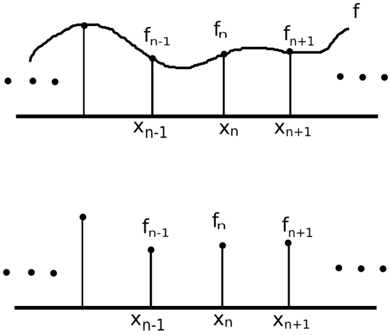


Figure : Sampling of a 1D signal.

- In the absence of the original signal f, we assume that the original signal lies in the span of the following *piece-wise linear* basis:

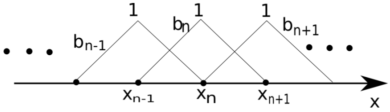


Figure : Piece-wise linear basis, note that $b_n(x) = 0, x \notin [x_{n-1}, x_{n+1}]$.

- Thus
  $\tilde{f}(x) = \ldots + a_{n-1}b_{n-1}(x) + a_n b_n(x) + a_{n+1}b_{n+1}(x) + \ldots$.
- Coefficients a_i's can be found by requiring that $\tilde{f}(x = x_n) = f_n$, the given sample values.
- For this particular choice of basis it is clear that $a_n = f_n, \forall n$, which gives for any $x \in [x_n, x_{n+1}]$,

$$\tilde{f}(x) = \frac{x_{n+1} - x}{x_{n+1} - x_n} f_n + \frac{x - x_n}{x_{n+1} - x_n} f_{n+1}.$$

- In the absence of the original signal f, we assume that the original signal lies in the span of the following *piece-wise linear* basis:

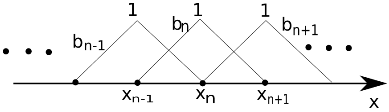


Figure : Piece-wise linear basis, note that $b_n(x) = 0, x \notin [x_{n-1}, x_{n+1}]$.

- Thus
$\tilde{f}(x) = \ldots + a_{n-1}b_{n-1}(x) + a_n b_n(x) + a_{n+1}b_{n+1}(x) + \ldots$.
- Coefficients a_i's can be found by requiring that $\tilde{f}(x = x_n) = f_n$, the given sample values.
- For this particular choice of basis it is clear that $a_n = f_n, \forall n$, which gives for any $x \in [x_n, x_{n+1}]$,

$$\tilde{f}(x) = \frac{x_{n+1} - x}{x_{n+1} - x_n} f_n + \frac{x - x_n}{x_{n+1} - x_n} f_{n+1}.$$

- In the absence of the original signal f, we assume that the original signal lies in the span of the following *piece-wise linear* basis:

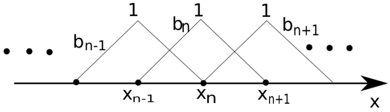


Figure : Piece-wise linear basis, note that $b_n(x) = 0, x \notin [x_{n-1}, x_{n+1}]$.

- Thus $\tilde{f}(x) = \ldots + a_{n-1}b_{n-1}(x) + a_n b_n(x) + a_{n+1}b_{n+1}(x) + \ldots$.
- Coefficients a_i's can be found by requiring that $\tilde{f}(x = x_n) = f_n$, the given sample values.
- For this particular choice of basis it is clear that $a_n = f_n, \forall n$, which gives for any $x \in [x_n, x_{n+1}]$,

$$\tilde{f}(x) = \frac{x_{n+1} - x}{x_{n+1} - x_n} f_n + \frac{x - x_n}{x_{n+1} - x_n} f_{n+1}.$$

- In the absence of the original signal f, we assume that the original signal lies in the span of the following *piece-wise linear* basis:

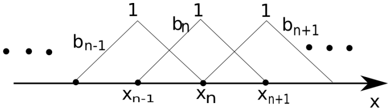


Figure : Piece-wise linear basis, note that $b_n(x) = 0, x \notin [x_{n-1}, x_{n+1}]$.

- Thus $\tilde{f}(x) = \ldots + a_{n-1}b_{n-1}(x) + a_n b_n(x) + a_{n+1}b_{n+1}(x) + \ldots$.
- Coefficients a_i's can be found by requiring that $\tilde{f}(x = x_n) = f_n$, the given sample values.
- For this particular choice of basis it is clear that $a_n = f_n, \forall n$, which gives for any $x \in [x_n, x_{n+1}]$,

$$\tilde{f}(x) = \frac{x_{n+1} - x}{x_{n+1} - x_n} f_n + \frac{x - x_n}{x_{n+1} - x_n} f_{n+1}.$$

- As should be clear from the equation, $\tilde{f}$ is simply a linear interpolation of the sample values.

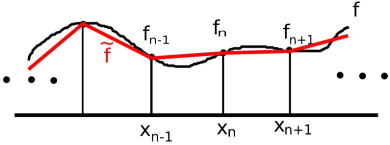


Figure : Original signal f and interpolated signal $\tilde{f}$.

- This allows us to compute derivatives of $\tilde{f}$ at any interior point $x \in (x_n, x_{n+1})$ as

$$\frac{d}{dx}\tilde{f}(x) = f_n \frac{d}{dx} b_n(x) + f_{n+1} \frac{d}{dx} b_{n+1}(x)$$

- As should be clear from the equation, $\tilde{f}$ is simply a linear interpolation of the sample values.

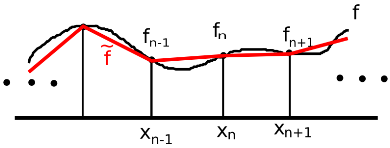


Figure : Original signal f and interpolated signal $\tilde{f}$.

- This allows us to compute derivatives of $\tilde{f}$ at any interior point $x \in (x_n, x_{n+1})$ as

$$\frac{d}{dx}\tilde{f}(x) = f_n \frac{d}{dx} b_n(x) + f_{n+1} \frac{d}{dx} b_{n+1}(x)$$

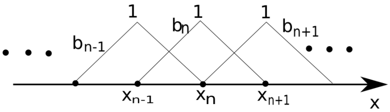


Figure : Piece-wise linear basis

- Obviously, for $x \in (x_n, x_{n+1})$,

$$\frac{d}{dx} b_n(x) = \frac{-1}{x_{n+1} - x_n} \text{ and } \frac{d}{dx} b_{n+1}(x) = \frac{1}{x_{n+1} - x_n}$$

- Thus for $x \in (x_n, x_{n+1})$,

$$\frac{d}{dx} \tilde{f}(x) = \frac{f_{n+1} - f_n}{x_{n+1} - x_n}$$

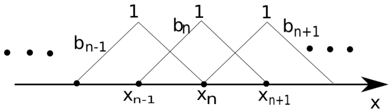


Figure : Piece-wise linear basis

- Obviously, for $x \in (x_n, x_{n+1})$,

$$\frac{d}{dx} b_n(x) = \frac{-1}{x_{n+1} - x_n} \text{ and } \frac{d}{dx} b_{n+1}(x) = \frac{1}{x_{n+1} - x_n}$$

- Thus for $x \in (x_n, x_{n+1})$,

$$\frac{d}{dx} \tilde{f}(x) = \frac{f_{n+1} - f_n}{x_{n+1} - x_n}$$

- At $x = x_n$?
- Several choices:
  - Forward Difference: $D_+\tilde{f}(x_n) = \frac{f_{n+1} - f_n}{x_{n+1} - x_n}$.
  - Backward Difference: $D_-\tilde{f}(x_n) = \frac{f_n - f_{n-1}}{x_n - x_{n-1}}$.
  - Any convex combination: $D_\omega\tilde{f}(x_n) = \omega D_+\tilde{f}(x_n) + (1-\omega)D_-\tilde{f}(x_n)$, where $\omega \in [0, 1]$.
  - In other words, compute an *average* of the two.

- At $x = x_n$?
- Several choices:
  - Forward Difference: $D_+\tilde{f}(x_n) = \frac{f_{n+1}-f_n}{x_{n+1}-x_n}$.
  - Backward Difference: $D_-\tilde{f}(x_n) = \frac{f_n - f_{n-1}}{x_n - x_{n-1}}$.
  - Any convex combination: $D_\omega\tilde{f}(x_n) = \omega D_+\tilde{f}(x_n) + (1-\omega)D_-\tilde{f}(x_n)$, where $\omega \in [0,1]$.
  - In other words, compute an *average* of the two.

- At $x = x_n$?
- Several choices:
  - Forward Difference: $D_+\tilde{f}(x_n) = \frac{f_{n+1}-f_n}{x_{n+1}-x_n}$.
  - Backward Difference: $D_-\tilde{f}(x_n) = \frac{f_n-f_{n-1}}{x_n-x_{n-1}}$.
  - Any convex combination: $D_\omega\tilde{f}(x_n) = \omega D_+\tilde{f}(x_n) + (1-\omega)D_-\tilde{f}(x_n)$, where $\omega \in [0,1]$.
  - In other words, compute an *average* of the two.

- At $x = x_n$?
- Several choices:
  - Forward Difference: $D_+\tilde{f}(x_n) = \frac{f_{n+1} - f_n}{x_{n+1} - x_n}$.
  - Backward Difference: $D_-\tilde{f}(x_n) = \frac{f_n - f_{n-1}}{x_n - x_{n-1}}$.
  - Any convex combination: $D_\omega \tilde{f}(x_n) = \omega D_+\tilde{f}(x_n) + (1-\omega) D_-\tilde{f}(x_n)$, where $\omega \in [0, 1]$.
  - In other words, compute an *average* of the two.

- At $x = x_n$?
- Several choices:
  - Forward Difference: $D_+\tilde{f}(x_n) = \frac{f_{n+1}-f_n}{x_{n+1}-x_n}$.
  - Backward Difference: $D_-\tilde{f}(x_n) = \frac{f_n-f_{n-1}}{x_n-x_{n-1}}$.
  - Any convex combination: $D_\omega\tilde{f}(x_n) = \omega D_+\tilde{f}(x_n) + (1-\omega)D_-\tilde{f}(x_n)$, where $\omega \in [0,1]$.
  - In other words, compute an *average* of the two.

- At $x = x_n$?
- Several choices:
  - Forward Difference: $D_+\tilde{f}(x_n) = \frac{f_{n+1} - f_n}{x_{n+1} - x_n}$.
  - Backward Difference: $D_-\tilde{f}(x_n) = \frac{f_n - f_{n-1}}{x_n - x_{n-1}}$.
  - Any convex combination:
    $D_\omega \tilde{f}(x_n) = \omega D_+\tilde{f}(x_n) + (1-\omega)D_-\tilde{f}(x_n)$, where $\omega \in [0, 1]$.
  - In other words, compute an *average* of the two.

Gradients of functions on a mesh

- Linear basis on a triangle of the mesh:

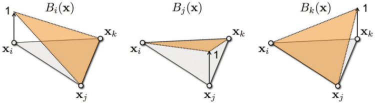


Figure : Three piece-wise linear basis per triangle. Image from BKPAL10.

- Any piece-wise linear function on the mesh with values at the vertices f_i can be written as

$$f(x) = f_i B_i(x) + f_j B_j(x) + f_k B_k(x)$$

  - The gradient of f at any $x \in Int(tri(x_i, x_j, x_k))$ is then

$$\nabla f(x) = f_i \nabla B_i(x) + f_j \nabla B_j(x) + f_k \nabla B_k(x)$$

Gradients of functions on a mesh

- Linear basis on a triangle of the mesh:

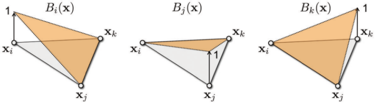


Figure : Three piece-wise linear basis per triangle. Image from BKPAL10.

- Any piece-wise linear function on the mesh with values at the vertices f_i can be written as

$$f(x) = f_i B_i(x) + f_j B_j(x) + f_k B_k(x)$$

- The gradient of f at any $x \in Int(tri(x_i, x_j, x_k))$ is then

$$\nabla f(x) = f_i \nabla B_i(x) + f_j \nabla B_j(x) + f_k \nabla B_k(x)$$

Gradients of functions on a mesh

- Linear basis on a triangle of the mesh:

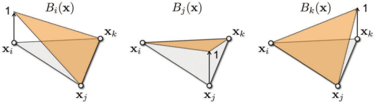


Figure : Three piece-wise linear basis per triangle. Image from BKPAL10.

- Any piece-wise linear function on the mesh with values at the vertices f_i can be written as

$$f(x) = f_i B_i(x) + f_j B_j(x) + f_k B_k(x)$$

- The gradient of f at any $x \in Int(tri(x_i, x_j, x_k))$ is then

$$\nabla f(x) = f_i \nabla B_i(x) + f_j \nabla B_j(x) + f_k \nabla B_k(x)$$

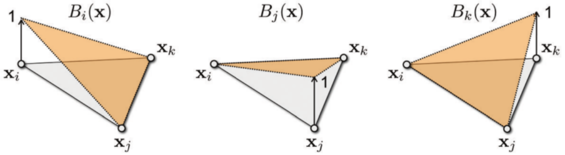


Figure : Three piece-wise linear basis per triangle. Image from BKPAL10.

- What is $\nabla B_i(x)$?
  - $\nabla B_i(x) = \frac{(x_k - x_j)^{\perp}}{2A_T}$.
- Note that $B_i(x) + B_j(x) + B_k(x) = 1$ gives $\nabla B_i(x) = -(\nabla B_j(x) + \nabla B_k(x))$.
  - Putting it together, we get

$$\nabla f(x) = (f_j - f_i)\nabla B_j(x) + (f_k - f_i)\nabla B_k(x)$$

- Note that $\nabla B_i(x) = \frac{N \times (x_k - x_j)}{2A_T}$, which gives

$$\nabla f(x) = f_i \frac{N \times (x_k - x_j)}{2A_T} + f_j \frac{N \times (x_i - x_k)}{2A_T} + f_k \frac{N \times (x_j - x_i)}{2A_T}$$

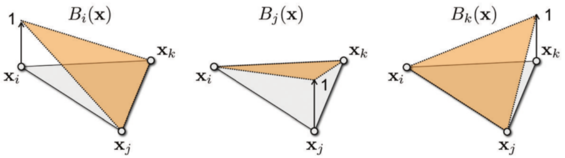


Figure : Three piece-wise linear basis per triangle. Image from BKPAL10.

- What is $\nabla B_i(x)$?
  - $\nabla B_i(x) = \frac{(x_k - x_j)^\perp}{2A_T}$.
- Note that $B_i(x) + B_j(x) + B_k(x) = 1$ gives $\nabla B_i(x) = -(\nabla B_j(x) + \nabla B_k(x))$.
  - Putting it together, we get

$$\nabla f(x) = (f_j - f_i)\nabla B_j(x) + (f_k - f_i)\nabla B_k(x)$$

- Note that $\nabla B_i(x) = \frac{N\times(x_k - x_j)}{2A_T}$, which gives

$$\nabla f(x) = f_i\frac{N \times (x_k - x_j)}{2A_T} + f_j\frac{N \times (x_i - x_k)}{2A_T} + f_k\frac{N \times (x_j - x_i)}{2A_T}$$

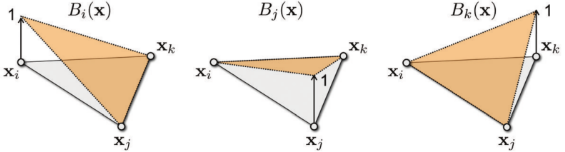


Figure : Three piece-wise linear basis per triangle. Image from BKPAL10.

- What is $\nabla B_i(x)$?
  - $\nabla B_i(x) = \frac{(x_k - x_j)^{\perp}}{2A_T}$.
- Note that $B_i(x) + B_j(x) + B_k(x) = 1$ gives $\nabla B_i(x) = -(\nabla B_j(x) + \nabla B_k(x))$.
  - Putting it together, we get

$$\nabla f(x) = (f_j - f_i)\nabla B_j(x) + (f_k - f_i)\nabla B_k(x)$$

- Note that $\nabla B_i(x) = \frac{N \times (x_k - x_j)}{2A_T}$, which gives

$$\nabla f(x) = f_i \frac{N \times (x_k - x_j)}{2A_T} + f_j \frac{N \times (x_i - x_k)}{2A_T} + f_k \frac{N \times (x_j - x_i)}{2A_T}$$

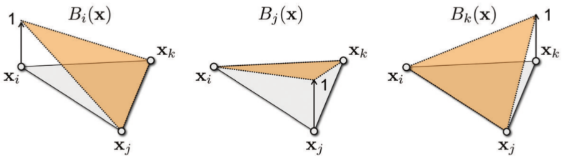


Figure : Three piece-wise linear basis per triangle. Image from BKPAL10.

- What is $\nabla B_i(x)$?
  - $\nabla B_i(x) = \frac{(x_k - x_j)^\perp}{2A_T}$.
- Note that $B_i(x) + B_j(x) + B_k(x) = 1$ gives $\nabla B_i(x) = -(\nabla B_j(x) + \nabla B_k(x))$.
  - Putting it together, we get

$$\nabla f(x) = (f_j - f_i)\nabla B_j(x) + (f_k - f_i)\nabla B_k(x)$$

- Note that $\nabla B_i(x) = \frac{N \times (x_k - x_j)}{2A_T}$, which gives

$$\nabla f(x) = f_i \frac{N \times (x_k - x_j)}{2A_T} + f_j \frac{N \times (x_i - x_k)}{2A_T} + f_k \frac{N \times (x_j - x_i)}{2A_T}$$

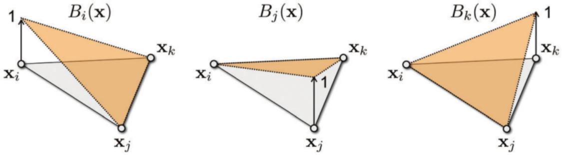


Figure : Three piece-wise linear basis per triangle. Image from BKPAL10.

- What is $\nabla B_i(x)$?
- $\nabla B_i(x) = \frac{(x_k - x_j)^{\perp}}{2A_T}$.
- Note that $B_i(x) + B_j(x) + B_k(x) = 1$ gives $\nabla B_i(x) = -(\nabla B_j(x) + \nabla B_k(x))$.
- Putting it together, we get

$$\nabla f(x) = (f_j - f_i)\nabla B_j(x) + (f_k - f_i)\nabla B_k(x)$$

- Note that $\nabla B_i(x) = \frac{N \times (x_k - x_j)}{2A_T}$, which gives

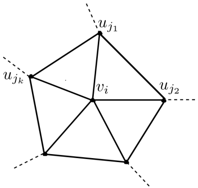
$$\nabla f(x) = f_i \frac{N \times (x_k - x_j)}{2A_T} + f_j \frac{N \times (x_i - x_k)}{2A_T} + f_k \frac{N \times (x_j - x_i)}{2A_T}$$

Surface Normal

- Surface normal for a triangle T with vertices (x_i, x_j, x_k), given in AC order seen from outside of the surface:
$N(T) = \frac{(x_j - x_i)\times(x_k - x_i)}{||(x_j - x_i)\times(x_k - x_i)||}$.
- On a vertex v: Weighted average of unit normals of all incident triangles.

$$N(v) = \frac{\sum_{T\in N_1(v)} a_T N(T)}{||\sum_{T\in N_1(v)} a_T N(T)||},$$

where a_T's are non-negative real scalars.
- *One-Ring*: Set of triangles incident to a given vertex v is called the one-ring of v and is denoted by $N_1(v)$.



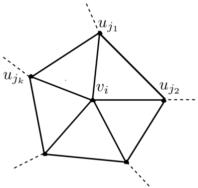
Surface Normal

- Surface normal for a triangle T with vertices (x_i, x_j, x_k), given in AC order seen from outside of the surface: $N(T) = \frac{(x_j - x_i) \times (x_k - x_i)}{||(x_j - x_i) \times (x_k - x_i)||}$.
- On a vertex v: Weighted average of unit normals of all incident triangles.

$$N(v) = \frac{\sum_{T \in N_1(v)} a_T N(T)}{|| \sum_{T \in N_1(v)} a_T N(T) ||},$$

where a_T's are non-negative real scalars.

- *One-Ring*: Set of triangles incident to a given vertex v is called the one-ring of v and is denoted by $N_1(v)$.

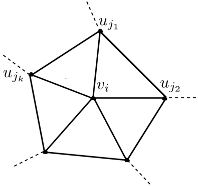


Surface Normal

- Surface normal for a triangle T with vertices (x_i, x_j, x_k), given in AC order seen from outside of the surface: $N(T) = \frac{(x_j - x_i) \times (x_k - x_i)}{||(x_j - x_i) \times (x_k - x_i)||}$.
- On a vertex v: Weighted average of unit normals of all incident triangles.

$$N(v) = \frac{\sum_{T \in N_1(v)} a_T N(T)}{|| \sum_{T \in N_1(v)} a_T N(T)||},$$

  where a_T's are non-negative real scalars.
- *One-Ring*: Set of triangles incident to a given vertex v is called the one-ring of v and is denoted by $N_1(v)$.



Laplacian

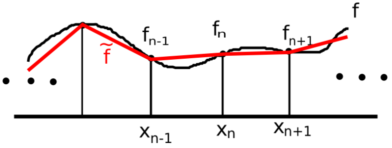


Figure : Original signal f and interpolated signal $\tilde{f}$.

- For now let's assume $x_n = \Delta n$, where Δ is a uniform sampling interval.
- $\frac{d^2\tilde{f}}{dx^2}(x) = 0, \;\; \forall x \in (x_n, x_{n+1})$.
- $\frac{d^2\tilde{f}}{dx^2}(x_n) = \frac{\frac{d\tilde{f}}{dx}\left(x_{n+\frac{1}{2}}\right) - \frac{d\tilde{f}}{dx}\left(x_{n-\frac{1}{2}}\right)}{\Delta} = \frac{\frac{f_{n+1}-f_n}{\Delta} - \frac{f_n - f_{n-1}}{\Delta}}{\Delta}$.
  - $\frac{d^2\tilde{f}}{dx^2}(x_n) = \frac{f_{n+1}+f_{n-1}-2f_n}{\Delta^2}$.

Laplacian

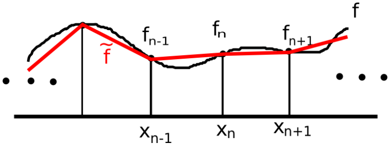


Figure : Original signal f and interpolated signal $\tilde{f}$.

- For now let's assume $x_n = \Delta n$, where Δ is a uniform sampling interval.
- $\frac{d^2\tilde{f}}{dx^2}(x) = 0, \ \forall x \in (x_n, x_{n+1})$.
- $\frac{d^2\tilde{f}}{dx^2}(x_n) = \frac{\frac{d\tilde{f}}{dx}\left(x_{n+\frac{1}{2}}\right) - \frac{d\tilde{f}}{dx}\left(x_{n-\frac{1}{2}}\right)}{\Delta} = \frac{\frac{f_{n+1}-f_n}{\Delta} - \frac{f_n - f_{n-1}}{\Delta}}{\Delta}$.
  - $\frac{d^2\tilde{f}}{dx^2}(x_n) = \frac{f_{n+1}+f_{n-1}-2f_n}{\Delta^2}$.

Laplacian

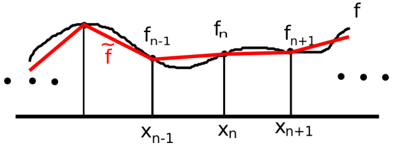


Figure : Original signal f and interpolated signal $\tilde{f}$.

- For now let's assume $x_n = \Delta n$, where Δ is a uniform sampling interval.
- $\frac{d^2\tilde{f}}{dx^2}(x) = 0, \ \forall x \in (x_n, x_{n+1})$.
- $\frac{d^2\tilde{f}}{dx^2}(x_n) = \frac{\frac{d\tilde{f}}{dx}\left(x_{n+\frac{1}{2}}\right) - \frac{d\tilde{f}}{dx}\left(x_{n-\frac{1}{2}}\right)}{\Delta} = \frac{\frac{f_{n+1}-f_n}{\Delta} - \frac{f_n - f_{n-1}}{\Delta}}{\Delta}$.
  - $\frac{d^2\tilde{f}}{dx^2}(x_n) = \frac{f_{n+1}+f_{n-1}-2f_n}{\Delta^2}$.

Laplacian

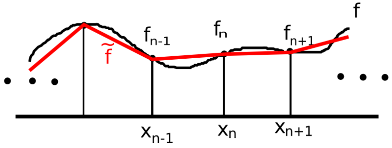


Figure : Original signal f and interpolated signal $\tilde{f}$.

- For now let's assume $x_n = \Delta n$, where Δ is a uniform sampling interval.
- $\frac{d^2\tilde{f}}{dx^2}(x) = 0, \ \forall x \in (x_n, x_{n+1})$.
- $\frac{d^2\tilde{f}}{dx^2}(x_n) = \frac{\frac{d\tilde{f}}{dx}\left(x_{n+\frac{1}{2}}\right) - \frac{d\tilde{f}}{dx}\left(x_{n-\frac{1}{2}}\right)}{\Delta} = \frac{\frac{f_{n+1}-f_n}{\Delta} - \frac{f_n - f_{n-1}}{\Delta}}{\Delta}$.
  - $\frac{d^2\tilde{f}}{dx^2}(x_n) = \frac{f_{n+1}+f_{n-1}-2f_n}{\Delta^2}$.

- $\frac{d^2\tilde{f}}{dx^2}(x_n) = \frac{f_{n+1}-f_n}{\Delta^2} + \frac{f_{n-1}-f_n}{\Delta^2}$

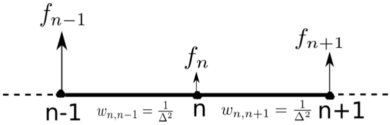


Figure : Signal domain interpreted as a graph.

- Denoting the Laplacian operator by L, we have

$$(L\tilde{f})_n = \sum_{j \in N(n)} w_{jn}(f_j - f_n), \quad w_{jn} = \frac{1}{\Delta^2}.$$

  - Define $w_{jn} = \frac{1}{Delta^2}$ if $j \in N(n)$, else $w_{jn} = 0$. We get

$$(L\tilde{f})_n = \sum_j w_{jn}(f_j - f_n).$$

- $\frac{d^2\tilde{f}}{dx^2}(x_n) = \frac{f_{n+1} - f_n}{\Delta^2} + \frac{f_{n-1} - f_n}{\Delta^2}$

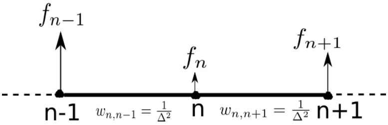


Figure : Signal domain interpreted as a graph.

- Denoting the Laplacian operator by L, we have

$$(L\tilde{f})_n = \sum_{j \in N(n)} w_{jn}(f_j - f_n), \quad w_{jn} = \frac{1}{\Delta^2}.$$

- Define $w_{jn} = \frac{1}{Delta^2}$ if $j \in N(n)$, else $w_{jn} = 0$. We get

$$(L\tilde{f})_n = \sum_j w_{jn}(f_j - f_n).$$

- $\frac{d^2\tilde{f}}{dx^2}(x_n) = \frac{f_{n+1}-f_n}{\Delta^2} + \frac{f_{n-1}-f_n}{\Delta^2}$

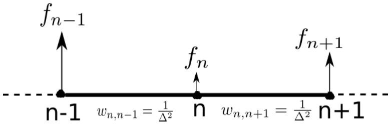


Figure : Signal domain interpreted as a graph.

- Denoting the Laplacian operator by L, we have

$$(L\tilde{f})_n = \sum_{j\in N(n)} w_{jn}(f_j - f_n), \quad w_{jn} = \frac{1}{\Delta^2}.$$

- Define $w_{jn} = \frac{1}{Delta^2}$ if $j \in N(n)$, else $w_{jn} = 0$. We get

$$(L\tilde{f})_n = \sum_j w_{jn}(f_j - f_n).$$

- As a matrix, we get

$$\begin{bmatrix} | \\ (L\tilde{f})_n \\ | \end{bmatrix} = \begin{bmatrix} - & w_{nn-1} & \overset{|}{\underset{|}{w_{nn}}} & w_{nn+1} & - \end{bmatrix} \begin{bmatrix} | \\ f_{n-1} \\ f_n \\ f_{n+1} \\ | \end{bmatrix},$$

where $w_{n,n} = -\sum_j w_{jn}$.

- The matrix A such that $A(i,j) = w_{ij}$ with $A(i,i) = 0$ is called the *Adjacency matrix* of the graph, while the diagonal matrix D with $D(i,i) = \sum_j A(i,j)$ is called the *Degree matrix* of the graph.
- Thus $L = A - D$.
- In graph theory, $\tilde{L} = D - A$ is called the *Graph Laplacian*.

- As a matrix, we get

$$\begin{bmatrix} | \\ (L\tilde{f})_n \\ | \end{bmatrix} = \begin{bmatrix} - & w_{nn-1} & \overset{|}{\underset{|}{w_{nn}}} & w_{nn+1} & - \end{bmatrix} \begin{bmatrix} | \\ f_{n-1} \\ f_n \\ f_{n+1} \\ | \end{bmatrix},$$

where $w_{n,n} = -\sum_j w_{jn}$.

- The matrix A such that $A(i,j) = w_{ij}$ with $A(i,i) = 0$ is called the *Adjacency matrix* of the graph, while the diagonal matrix D with $D(i,i) = \sum_j A(i,j)$ is called the *Degree matrix* of the graph.
- Thus $L = A - D$.
- In graph theory, $\tilde{L} = D - A$ is called the *Graph Laplacian*.

- As a matrix, we get

$$\begin{bmatrix} | \\ (L\tilde{f})_n \\ | \end{bmatrix} = \begin{bmatrix} - & w_{nn-1} & \overset{|}{w_{nn}} & w_{nn+1} & - \\ & & | & & \end{bmatrix} \begin{bmatrix} | \\ f_{n-1} \\ f_n \\ f_{n+1} \\ | \end{bmatrix},$$

  where $w_{n,n} = -\sum_j w_{jn}$.

- The matrix A such that $A(i,j) = w_{ij}$ with $A(i,i) = 0$ is called the *Adjacency matrix* of the graph, while the diagonal matrix D with $D(i,i) = \sum_j A(i,j)$ is called the *Degree matrix* of the graph.
- Thus $L = A - D$.
- In graph theory, $\tilde{L} = D - A$ is called the *Graph Laplacian*.

- As a matrix, we get

$$\begin{bmatrix} | \\ (L\tilde{f})_n \\ | \end{bmatrix} = \begin{bmatrix} - & w_{nn-1} & \overset{|}{\underset{|}{w_{nn}}} & w_{nn+1} & - \end{bmatrix} \begin{bmatrix} | \\ f_{n-1} \\ f_n \\ f_{n+1} \\ | \end{bmatrix},$$

where $w_{n,n} = -\sum_j w_{jn}$.

- The matrix A such that $A(i,j) = w_{ij}$ with $A(i,i) = 0$ is called the *Adjacency matrix* of the graph, while the diagonal matrix D with $D(i,i) = \sum_j A(i,j)$ is called the *Degree matrix* of the graph.
- Thus $L = A - D$.
- In graph theory, $\tilde{L} = D - A$ is called the *Graph Laplacian*.

Laplacian for meshes

- A mesh is a graph; note $\Delta = d(x_n, x_{n+1})$.

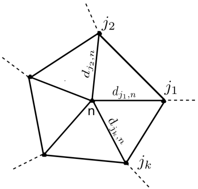


Figure : Mesh as a graph, $d_{jn} = ||x_j - x_n||^2$

.

- If vertex i and j are edge-connected $A(i,j) = \frac{1}{d_{i,j}}$, else $A(i,j) = 0$.
  - Construct the degree matrix D with $D(i,i) = \sum_j A(i,j)$, and the Laplacian $L = A - D$.

Laplacian for meshes

- A mesh is a graph; note $\Delta = d(x_n, x_{n+1})$.

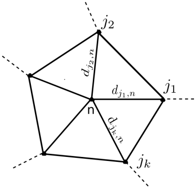


Figure : Mesh as a graph, $d_{jn} = ||x_j - x_n||^2$

.

- If vertex i and j are edge-connected $A(i,j) = \frac{1}{d_{i,j}}$, else $A(i,j) = 0$.
  - Construct the degree matrix D with $D(i,i) = \sum_j A(i,j)$, and the Laplacian $L = A - D$.

Laplacian for meshes

- A mesh is a graph; note $\Delta = d(x_n, x_{n+1})$.

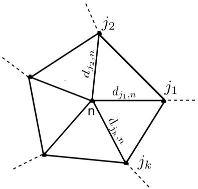


Figure : Mesh as a graph, $d_{jn} = ||x_j - x_n||^2$

.

- If vertex i and j are edge-connected $A(i,j) = \frac{1}{d_{i,j}}$, else $A(i,j) = 0$.
  - Construct the degree matrix D with $D(i,i) = \sum_j A(i,j)$, and the Laplacian $L = A - D$.

Thumb rules for discrete estimation

- Properties of the measure in the continuous setting should be satisfied by the discrete estimate.
- Order of error: Finer sampling $\Rightarrow$ Error should go down.
- Estimate should be robust. Robustness $\Rightarrow$ average (a.k.a low pass filter).

Thumb rules for discrete estimation

- Properties of the measure in the continuous setting should be satisfied by the discrete estimate.
- Order of error: Finer sampling $\Rightarrow$ Error should go down.
- Estimate should be robust. Robustness $\rightarrow$ average (a.k.a low pass filter).

Thumb rules for discrete estimation

- Properties of the measure in the continuous setting should be satisfied by the discrete estimate.
- Order of error: Finer sampling $\Rightarrow$ Error should go down.
- Estimate should be robust. Robustness $\Rightarrow$ average (a.k.a low pass filter).

Thumb rules for discrete estimation

- Properties of the measure in the continuous setting should be satisfied by the discrete estimate.
- Order of error: Finer sampling $\Rightarrow$ Error should go down.
- Estimate should be robust. Robustness $\rightarrow$ average (a.k.a low pass filter).

Thumb rules for discrete estimation

- Properties of the measure in the continuous setting should be satisfied by the discrete estimate.
- Order of error: Finer sampling $\Rightarrow$ Error should go down.
- Estimate should be robust. Robustness $\rightarrow$ average (a.k.a low pass filter).

Thumb rules for discrete estimation

- Properties of the measure in the continuous setting should be satisfied by the discrete estimate.
- Order of error: Finer sampling $\Rightarrow$ Error should go down.
- Estimate should be robust. Robustness $\rightarrow$ average (a.k.a low pass filter).

Laplace-Beltrami Operator (LBO)

- **Definition 1:** $\Delta := div \circ grad$.
- In $\mathbb{R}^n$, $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \ldots + \frac{\partial^2}{\partial x_n^2}$, LBO = Laplacian.
- Properties/Characteristics of LBO:
  - We know that $A(\sigma) = \int\int \sqrt{EG - F^2}\, da$, for $\sigma : \Omega \subset \mathbb{R}^2 \to S \subset \mathbb{R}^3$.
  - For $\sigma(u, v) = (u, v, f(u, v))$, $EG - F^2 = 1 + f_u^2 + f_v^2$.
  - Characterize an σ^* such that $A(\sigma^*)$ is minimum, given a fixed boundary $\sigma|_{\partial\Omega} = \Gamma$.

Laplace-Beltrami Operator (LBO)

- **Definition 1:** $\Delta := div \circ grad$.
- In $\mathbb{R}^n$, $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \ldots + \frac{\partial^2}{\partial x_n^2}$, LBO $=$ Laplacian.
- Properties/Characteristics of LBO:
  - We know that $A(\sigma) = \int \int \sqrt{EG - F^2} \, da$, for $\sigma : \Omega \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$.
  - For $\sigma(u, v) = (u, v, f(u, v))$, $EG - F^2 = 1 + f_u^2 + f_v^2$.
  - Characterize an σ^* such that $A(\sigma^*)$ is minimum, given a fixed boundary $\sigma|_{\partial\Omega} = \Gamma$.

Laplace-Beltrami Operator (LBO)

- **Definition 1:** $\Delta := div \circ grad$.
- In $\mathbb{R}^n$, $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \ldots + \frac{\partial^2}{\partial x_n^2}$, LBO = Laplacian.
- Properties/Characteristics of LBO:
  - We know that $A(\sigma) = \int\int \sqrt{EG - F^2}\, da$, for $\sigma : \Omega \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$.
  - For $\sigma(u, v) = (u, v, f(u, v))$, $EG - F^2 = 1 + f_u^2 + f_v^2$.
  - Characterize an σ^* such that $A(\sigma^*)$ is minimum, given a fixed boundary $\sigma|_{\partial\Omega} = \Gamma$.

Laplace-Beltrami Operator (LBO)

- **Definition 1:** $\Delta := div \circ grad$.
- In $\mathbb{R}^n$, $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \ldots + \frac{\partial^2}{\partial x_n^2}$, LBO = Laplacian.
- Properties/Characteristics of LBO:
  - We know that $A(\sigma) = \int \int \sqrt{EG - F^2}\, da$, for $\sigma : \Omega \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$.
  - For $\sigma(u, v) = (u, v, f(u, v))$, $EG - F^2 = 1 + f_u^2 + f_v^2$.
  - Characterize an σ^* such that $A(\sigma^*)$ is minimum, given a fixed boundary $\sigma|_{\partial\Omega} = \Gamma$.

Laplace-Beltrami Operator (LBO)

- **Definition 1:** $\Delta := div \circ grad$.
- In $\mathbb{R}^n$, $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \ldots + \frac{\partial^2}{\partial x_n^2}$, LBO = Laplacian.
- Properties/Characteristics of LBO:
  - We know that $A(\sigma) = \int\int \sqrt{EG - F^2}\, da$, for $\sigma : \Omega \subset \mathbb{R}^2 \to S \subset \mathbb{R}^3$.
  - For $\sigma(u, v) = (u, v, f(u, v))$, $EG - F^2 = 1 + f_u^2 + f_v^2$.
  - Characterize an σ^* such that $A(\sigma^*)$ is minimum, given a fixed boundary $\sigma|_{\partial\Omega} = \Gamma$.

Laplace-Beltrami Operator (LBO)

- **Definition 1:** $\Delta := div \circ grad$.
- In $\mathbb{R}^n$, $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \ldots + \frac{\partial^2}{\partial x_n^2}$, LBO $=$ Laplacian.
- Properties/Characteristics of LBO:
  - We know that $A(\sigma) = \int\int \sqrt{EG - F^2}\, da$, for $\sigma : \Omega \subset \mathbb{R}^2 \to S \subset \mathbb{R}^3$.
  - For $\sigma(u, v) = (u, v, f(u, v))$, $EG - F^2 = 1 + f_u^2 + f_v^2$.
  - Characterize an σ^* such that $A(\sigma^*)$ is minimum, given a fixed boundary $\sigma|_{\partial\Omega} = \Gamma$.

- Let $g : \Omega \to \mathbb{R}$ be a perturbation added to f by a scalar amount s, so that the surface σ becomes $\tilde{\sigma}_s = (u, v, f(u, v) + sg(u, v))$, with $g|_{\partial\Omega} = 0$.
- One can then define the Directional derivative of A at f in the direction g as $D_g A(f) := \frac{d}{ds} A(f + sg)|_{s=0}$.
  - Another notation for DD: $\frac{\partial A}{\partial g}(f)$.
  - As in finite dimensional cases, $\frac{\partial A}{\partial g}(f) = \langle \nabla A(f), g \rangle$.
  - It can be derived that

$$\nabla A(f) = -\frac{(1 + f_v^2) f_{uu} + (1 + f_u^2) f_{vv} - 2 f_u f_v f_{uv}}{(1 + f_u^2 + f_v^2)^{\frac{3}{2}}}.$$

  - Thus f^* represents a minimal surface if $\nabla A(f^*) = 0$.
  - Note $E = 1 + f_u^2, F = f_u f_v, G = 1 + f_v^2$.
  - Also $L = \langle \sigma_{uu}, n \rangle = \frac{f_{uu}}{\sqrt{1+f_u^2+f_v^2}}, M = \frac{f_{uv}}{\sqrt{1+f_u^2+f_v^2}}, N = \frac{f_{vv}}{\sqrt{1+f_u^2+f_v^2}}$.

- Let $g : \Omega \to \mathbb{R}$ be a perturbation added to f by a scalar amount s, so that the surface σ becomes $\tilde{\sigma}_s = (u, v, f(u, v) + sg(u, v))$, with $g|_{\partial\Omega} = 0$.
- One can then define the Directional derivative of A at f in the direction g as $D_g A(f) := \frac{d}{ds} A(f + sg)|_{s=0}$.
  - Another notation for DD: $\frac{\partial A}{\partial g}(f)$.
  - As in finite dimensional cases, $\frac{\partial A}{\partial g}(f) = \langle \nabla A(f), g \rangle$.
  - It can be derived that

$$\nabla A(f) = -\frac{(1 + f_v^2) f_{uu} + (1 + f_u^2) f_{vv} - 2 f_u f_v f_{uv}}{(1 + f_u^2 + f_v^2)^{\frac{3}{2}}}.$$

  - Thus f^* represents a minimal surface if $\nabla A(f^*) = 0$.
  - Note $E = 1 + f_u^2, F = f_u f_v, G = 1 + f_v^2$.
  - Also $L = \langle \sigma_{uu}, n \rangle = \frac{f_{uu}}{\sqrt{1+f_u^2+f_v^2}}, M = \frac{f_{uv}}{\sqrt{1+f_u^2+f_v^2}}, N = \frac{f_{vv}}{\sqrt{1+f_u^2+f_v^2}}$.

- Let $g : \Omega \to \mathbb{R}$ be a perturbation added to f by a scalar amount s, so that the surface σ becomes $\tilde{\sigma}_s = (u, v, f(u, v) + sg(u, v))$, with $g|_{\partial\Omega} = 0$.
- One can then define the Directional derivative of A at f in the direction g as $D_g A(f) := \frac{d}{ds} A(f + sg)|_{s=0}$.
  - Another notation for DD: $\frac{\partial A}{\partial g}(f)$.
  - As in finite dimensional cases, $\frac{\partial A}{\partial g}(f) = \langle \nabla A(f), g \rangle$.
  - It can be derived that

$$\nabla A(f) = -\frac{(1 + f_v^2) f_{uu} + (1 + f_u^2) f_{vv} - 2 f_u f_v f_{uv}}{(1 + f_u^2 + f_v^2)^{\frac{3}{2}}}.$$

  - Thus f^* represents a minimal surface if $\nabla A(f^*) = 0$.
  - Note $E = 1 + f_u^2, F = f_u f_v, G = 1 + f_v^2$.
  - Also $L = \langle \sigma_{uu}, n \rangle = \frac{f_{uu}}{\sqrt{1+f_u^2+f_v^2}}, M = \frac{f_{uv}}{\sqrt{1+f_u^2+f_v^2}}, N = \frac{f_{vv}}{\sqrt{1+f_u^2+f_v^2}}$.

- Let $g : \Omega \to \mathbb{R}$ be a perturbation added to f by a scalar amount s, so that the surface σ becomes $\tilde{\sigma_s} = (u, v, f(u, v) + sg(u, v))$, with $g|_{\partial\Omega} = 0$.
- One can then define the Directional derivative of A at f in the direction g as $D_g A(f) := \frac{d}{ds} A(f + sg)|_{s=0}$.
  - Another notation for DD: $\frac{\partial A}{\partial g}(f)$.
  - As in finite dimensional cases, $\frac{\partial A}{\partial g}(f) = \langle \nabla A(f), g \rangle$.
  - It can be derived that

$$\nabla A(f) = -\frac{(1 + f_v^2) f_{uu} + (1 + f_u^2) f_{vv} - 2 f_u f_v f_{uv}}{(1 + f_u^2 + f_v^2)^{\frac{3}{2}}}.$$

  - Thus f^* represents a minimal surface if $\nabla A(f^*) = 0$.
  - Note $E = 1 + f_u^2, F = f_u f_v, G = 1 + f_v^2$.
  - Also
    $L = \langle \sigma_{uu}, n \rangle = \frac{f_{uu}}{\sqrt{1+f_u^2+f_v^2}}, M = \frac{f_{uv}}{\sqrt{1+f_u^2+f_v^2}}, N = \frac{f_{vv}}{\sqrt{1+f_u^2+f_v^2}}.$

- Let $g : \Omega \to \mathbb{R}$ be a perturbation added to f by a scalar amount s, so that the surface σ becomes $\tilde{\sigma}_s = (u, v, f(u,v) + sg(u,v))$, with $g|_{\partial\Omega} = 0$.
- One can then define the Directional derivative of A at f in the direction g as $D_g A(f) := \frac{d}{ds} A(f + sg)|_{s=0}$.
  - Another notation for DD: $\frac{\partial A}{\partial g}(f)$.
  - As in finite dimensional cases, $\frac{\partial A}{\partial g}(f) = \langle \nabla A(f), g \rangle$.
  - It can be derived that

$$\nabla A(f) = -\frac{(1 + f_v^2) f_{uu} + (1 + f_u^2) f_{vv} - 2 f_u f_v f_{uv}}{(1 + f_u^2 + f_v^2)^{\frac{3}{2}}}.$$

  - Thus f^* represents a minimal surface if $\nabla A(f^*) = 0$.
  - Note $E = 1 + f_u^2, F = f_u f_v, G = 1 + f_v^2$.
  - Also
$L = \langle \sigma_{uu}, n \rangle = \frac{f_{uu}}{\sqrt{1+f_u^2+f_v^2}}, M = \frac{f_{uv}}{\sqrt{1+f_u^2+f_v^2}}, N = \frac{f_{vv}}{\sqrt{1+f_u^2+f_v^2}}$.

- Let $g : \Omega \to \mathbb{R}$ be a perturbation added to f by a scalar amount s, so that the surface σ becomes $\tilde{\sigma}_s = (u, v, f(u, v) + sg(u, v))$, with $g|_{\partial\Omega} = 0$.
- One can then define the Directional derivative of A at f in the direction g as $D_g A(f) := \frac{d}{ds} A(f + sg)|_{s=0}$.
  - Another notation for DD: $\frac{\partial A}{\partial g}(f)$.
  - As in finite dimensional cases, $\frac{\partial A}{\partial g}(f) = \langle \nabla A(f), g \rangle$.
  - It can be derived that

$$\nabla A(f) = -\frac{(1 + f_v^2) f_{uu} + (1 + f_u^2) f_{vv} - 2 f_u f_v f_{uv}}{(1 + f_u^2 + f_v^2)^{\frac{3}{2}}}.$$

  - Thus f^* represents a minimal surface if $\nabla A(f^*) = 0$.
  - Note $E = 1 + f_u^2, F = f_u f_v, G = 1 + f_v^2$.
  - Also $L = \langle \sigma_{uu}, n \rangle = \frac{f_{uu}}{\sqrt{1+f_u^2+f_v^2}}, M = \frac{f_{uv}}{\sqrt{1+f_u^2+f_v^2}}, N = \frac{f_{vv}}{\sqrt{1+f_u^2+f_v^2}}$.

- Let $g : \Omega \to \mathbb{R}$ be a perturbation added to f by a scalar amount s, so that the surface σ becomes $\tilde{\sigma}_s = (u, v, f(u, v) + sg(u, v))$, with $g|_{\partial\Omega} = 0$.
- One can then define the Directional derivative of A at f in the direction g as $D_g A(f) := \frac{d}{ds} A(f + sg)|_{s=0}$.
  - Another notation for DD: $\frac{\partial A}{\partial g}(f)$.
  - As in finite dimensional cases, $\frac{\partial A}{\partial g}(f) = \langle \nabla A(f), g \rangle$.
  - It can be derived that

$$\nabla A(f) = -\frac{(1 + f_v^2) f_{uu} + (1 + f_u^2) f_{vv} - 2 f_u f_v f_{uv}}{(1 + f_u^2 + f_v^2)^{\frac{3}{2}}}.$$

  - Thus f^* represents a minimal surface if $\nabla A(f^*) = 0$.
  - Note $E = 1 + f_u^2, F = f_u f_v, G = 1 + f_v^2$.
  - Also $L = \langle \sigma_{uu}, n \rangle = \frac{f_{uu}}{\sqrt{1+f_u^2+f_v^2}}, M = \frac{f_{uv}}{\sqrt{1+f_u^2+f_v^2}}, N = \frac{f_{vv}}{\sqrt{1+f_u^2+f_v^2}}$.

- Let $g : \Omega \to \mathbb{R}$ be a perturbation added to f by a scalar amount s, so that the surface σ becomes $\tilde{\sigma_s} = (u, v, f(u, v) + sg(u, v))$, with $g|_{\partial\Omega} = 0$.
- One can then define the Directional derivative of A at f in the direction g as $D_g A(f) := \frac{d}{ds} A(f + sg)|_{s=0}$.
- Another notation for DD: $\frac{\partial A}{\partial g}(f)$.
- As in finite dimensional cases, $\frac{\partial A}{\partial g}(f) = \langle \nabla A(f), g \rangle$.
- It can be derived that

$$\nabla A(f) = -\frac{(1 + f_v^2)f_{uu} + (1 + f_u^2)f_{vv} - 2f_u f_v f_{uv}}{(1 + f_u^2 + f_v^2)^{\frac{3}{2}}}.$$

- Thus f^* represents a minimal surface if $\nabla A(f^*) = 0$.
- Note $E = 1 + f_u^2, F = f_u f_v, G = 1 + f_v^2$.
- Also $L = \langle \sigma_{uu}, n \rangle = \frac{f_{uu}}{\sqrt{1+f_u^2+f_v^2}}, M = \frac{f_{uv}}{\sqrt{1+f_u^2+f_v^2}}, N = \frac{f_{vv}}{\sqrt{1+f_u^2+f_v^2}}.$

- Thus $\nabla A(f) = -\frac{GL+EN-2FM}{EG-F^2}$.
- Note $H = Tr([S]) = Tr(\mathcal{F}_1^{-1}\mathcal{F}_2) = \frac{GL+EN-2FM}{2(EG-F^2)}$.
  - Thus, $\nabla A(f) = -2H$.
- Given a surface σ (or f), how can one turn it into a minimal surface?
  - Gradient descent: $\frac{\partial f}{\partial t} = -\nabla A(f)$, or $\frac{\partial \sigma}{\partial t} = -\nabla A(f)\hat{z}$, where $\hat{z} = (0, 0, 1)$.

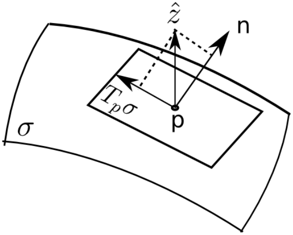


Figure : Area minimization surface evolution

- The famous Mean Curvature Flow: $\frac{\partial \sigma}{\partial t} = -2H\mathbf{n}$

- Thus $\nabla A(f) = -\frac{GL+EN-2FM}{EG-F^2}$.
- Note $H = Tr([S]) = Tr(\mathcal{F}_1^{-1}\mathcal{F}_2) = \frac{GL+EN-2FM}{2(EG-F^2)}$.
  - Thus, $\nabla A(f) = -2H$.
- Given a surface σ (or f), how can one turn it into a minimal surface?
  - Gradient descent: $\frac{\partial f}{\partial t} = -\nabla A(f)$, or $\frac{\partial \sigma}{\partial t} = -\nabla A(f)\hat{z}$, where $\hat{z} = (0,0,1)$.

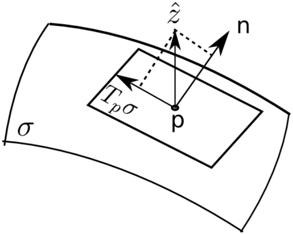


Figure : Area minimization surface evolution

- The famous Mean Curvature Flow: $\frac{\partial \sigma}{\partial t} = -2Hn$

- Thus $\nabla A(f) = -\frac{GL+EN-2FM}{EG-F^2}$.
- Note $H = Tr([S]) = Tr(\mathcal{F}_1^{-1}\mathcal{F}_2) = \frac{GL+EN-2FM}{2(EG-F^2)}$.
  - Thus, $\nabla A(f) = -2H$.
- Given a surface σ (or f), how can one turn it into a minimal surface?
  - Gradient descent: $\frac{\partial f}{\partial t} = -\nabla A(f)$, or $\frac{\partial \sigma}{\partial t} = -\nabla A(f)\hat{z}$, where $\hat{z} = (0,0,1)$.

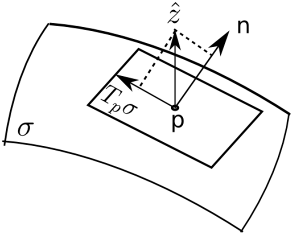


Figure : Area minimization surface evolution

- The famous Mean Curvature Flow: $\frac{\partial \sigma}{\partial t} = -2Hn$.

- Thus $\nabla A(f) = -\frac{GL+EN-2FM}{EG-F^2}$.
- Note $H = Tr([S]) = Tr(\mathcal{F}_1^{-1}\mathcal{F}_2) = \frac{GL+EN-2FM}{2(EG-F^2)}$.
  - Thus, $\nabla A(f) = -2H$.
- Given a surface σ (or f), how can one turn it into a minimal surface?
  - Gradient descent: $\frac{\partial f}{\partial t} = -\nabla A(f)$, or $\frac{\partial \sigma}{\partial t} = -\nabla A(f)\hat{z}$, where $\hat{z} = (0,0,1)$.

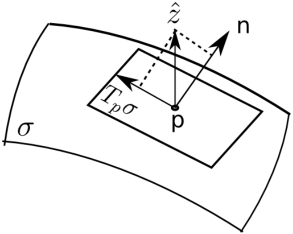


Figure: Area minimization surface evolution

- The famous Mean Curvature Flow

- Thus $\nabla A(f) = -\frac{GL+EN-2FM}{EG-F^2}$.
- Note $H = Tr([S]) = Tr(\mathcal{F}_1^{-1}\mathcal{F}_2) = \frac{GL+EN-2FM}{2(EG-F^2)}$.
  - Thus, $\nabla A(f) = -2H$.
- Given a surface σ (or f), how can one turn it into a minimal surface?
  - Gradient descent: $\frac{\partial f}{\partial t} = -\nabla A(f)$, or $\frac{\partial \sigma}{\partial t} = -\nabla A(f)\hat{z}$, where $\hat{z} = (0,0,1)$.

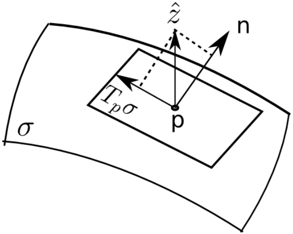


Figure : Area minimization surface evolution

- The famous Mean Curvature Flow: $\frac{\partial \sigma}{\partial t} = 2Hn$.

- Thus $\nabla A(f) = -\frac{GL+EN-2FM}{EG-F^2}$.
- Note $H = Tr([S]) = Tr(\mathcal{F}_1^{-1}\mathcal{F}_2) = \frac{GL+EN-2FM}{2(EG-F^2)}$.
  - Thus, $\nabla A(f) = -2H$.
- Given a surface σ (or f), how can one turn it into a minimal surface?
  - Gradient descent: $\frac{\partial f}{\partial t} = -\nabla A(f)$, or $\frac{\partial \sigma}{\partial t} = -\nabla A(f)\hat{z}$, where $\hat{z} = (0, 0, 1)$.

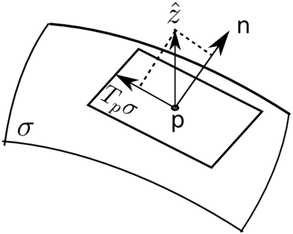


Figure : Area minimization surface evolution

- The famous Mean Curvature Flow: $\frac{\partial \sigma}{\partial t} = 2Hn$.

- Thus $\nabla A(f) = -\frac{GL+EN-2FM}{EG-F^2}$.
- Note $H = Tr([S]) = Tr(\mathcal{F}_1^{-1}\mathcal{F}_2) = \frac{GL+EN-2FM}{2(EG-F^2)}$.
  - Thus, $\nabla A(f) = -2H$.
- Given a surface σ (or f), how can one turn it into a minimal surface?
  - Gradient descent: $\frac{\partial f}{\partial t} = -\nabla A(f)$, or $\frac{\partial \sigma}{\partial t} = -\nabla A(f)\hat{z}$, where $\hat{z} = (0,0,1)$.

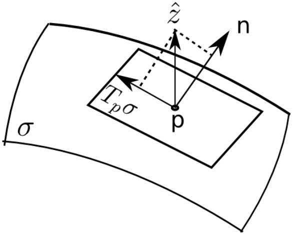


Figure : Area minimization surface evolution

- The famous **Mean Curvature Flow**: $\frac{\partial \sigma}{\partial t} = 2Hn$.

What has this got to do with LBO?

- *Isothermal or Conformal parameterization*: If $F = 0$ and $E = G$ at every point of the surface, we say that the parameters are *isothermal* or *conformal*.
  - Assuming such parameters[1] u, v exist and $E = G = 1$, $H = L + N$.
  - In this case, $\langle \sigma_u, \sigma_u \rangle = 1$, thus $\sigma_{uu} + \sigma_{vv} || n$.
  - Conformal parameterization $\Rightarrow \Delta\sigma = Hn$.
- **Definition 2**[2]: LBO is the operator that assigns to each point $p \in M$ the *mean curvature vector* $H(p)n(p)$.

[1] Gauss(1822) proved existence of local conformal parameterization for any 2D surface, while global existence was shown by Lichtenstein (1916).

[2] Meyer et al., Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, Intl. w'shop on Vis. & Math., 2002

What has this got to do with LBO?

- *Isothermal or Conformal parameterization*: If $F = 0$ and $E = G$ at every point of the surface, we say that the parameters are *isothermal* or *conformal*.
  - Assuming such parameters[1] u, v exist and $E = G = 1$, $H = L + N$.
  - In this case, $\langle \sigma_u, \sigma_u \rangle = 1$, thus $\sigma_{uu} + \sigma_{vv} || n$.
  - Conformal parameterization $\Rightarrow \Delta\sigma = Hn$.
- **Definition 2**[2]: LBO is the operator that assigns to each point $p \in M$ the *mean curvature vector* $H(p)n(p)$.

[1] Gauss(1822) proved existence of local conformal parameterization for any 2D surface, while global existence was shown by Lichtenstein (1916).

[2] Meyer et al., Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, Intl. w'shop on Vis. & Math., 2002

What has this got to do with LBO?

- *Isothermal or Conformal parameterization*: If $F = 0$ and $E = G$ at every point of the surface, we say that the parameters are *isothermal* or *conformal*.
  - Assuming such parameters[1] u, v exist and $E = G = 1$, $H = L + N$.
  - In this case, $\langle \sigma_u, \sigma_u \rangle = 1$, thus $\sigma_{uu} + \sigma_{vv} || n$.
  - Conformal parameterization $\Rightarrow \Delta\sigma = Hn$.
- **Definition 2**[2]: LBO is the operator that assigns to each point $p \in M$ the *mean curvature vector* $H(p)n(p)$.

[1] Gauss(1822) proved existence of local conformal parameterization for any 2D surface, while global existence was shown by Lichtenstein (1916).

[2] Meyer et al., Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, Intl. w'shop on Vis. & Math., 2002

What has this got to do with LBO?

- *Isothermal or Conformal parameterization*: If $F = 0$ and $E = G$ at every point of the surface, we say that the parameters are *isothermal* or *conformal*.
  - Assuming such parameters[1] u, v exist and $E = G = 1$, $H = L + N$.
  - In this case, $\langle \sigma_u, \sigma_u \rangle = 1$, thus $\sigma_{uu} + \sigma_{vv} || n$.
  - Conformal parameterization $\Rightarrow \Delta\sigma = Hn$.
- **Definition 2**[2]: LBO is the operator that assigns to each point $p \in M$ the *mean curvature vector* $H(p)n(p)$.

[1] Gauss(1822) proved existence of local conformal parameterization for any 2D surface, while global existence was shown by Lichtenstein (1916).

[2] Meyer et al., Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, Intl. w'shop on Vis. & Math., 2002

What has this got to do with LBO?

- *Isothermal or Conformal parameterization*: If $F = 0$ and $E = G$ at every point of the surface, we say that the parameters are *isothermal* or *conformal*.
  - Assuming such parameters[1] u, v exist and $E = G = 1$, $H = L + N$.
  - In this case, $\langle \sigma_u, \sigma_u \rangle = 1$, thus $\sigma_{uu} + \sigma_{vv} || n$.
  - Conformal parameterization $\Rightarrow \Delta\sigma = Hn$.
- **Definition 2**[2]: LBO is the operator that assigns to each point $p \in M$ the *mean curvature vector* $H(p)n(p)$.

[1] Gauss(1822) proved existence of local conformal parameterization for any 2D surface, while global existence was shown by Lichtenstein (1916).

[2] Meyer et al., Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, Intl. w'shop on Vis. & Math., 2002

Dirichlet Energy

- A parameterization σ of a surface maps points from $\mathbb{R}^2$ to points in $\mathbb{R}^3$.
- A measure of how rapidly σ varies:

$$E(\sigma) = \frac{1}{2}\int\int ||\frac{\partial\sigma}{\partial u}||^2 + ||\frac{\partial\sigma}{\partial v}||^2 \, du \, dv$$

- Typically written as $E(\sigma) = \frac{1}{2}\int\int ||\nabla\sigma||^2 \, du \, dv$ or $E(\sigma) = \frac{1}{2}\int\int Tr(J\sigma^t J\sigma) \, du \, dv$, and is called the *Dirichlet Energy*.
- With conformal parameterization $\sqrt{EG - F^2} = E$, thus $A(\sigma) = E(\sigma)$.
- The minimizer of $E(\sigma)$ also satisfies $\nabla A(\sigma) = Hn = \Delta\sigma = 0$.
- Functions that satisfy[3] $\Delta\sigma = 0$ are called *Harmonic functions*.

[3] This is the famous Laplace equation.

Dirichlet Energy

- A parameterization σ of a surface maps points from $\mathbb{R}^2$ to points in $\mathbb{R}^3$.
- A measure of how rapidly σ varies:

$$E(\sigma) = \frac{1}{2} \int \int ||\frac{\partial \sigma}{\partial u}||^2 + ||\frac{\partial \sigma}{\partial v}||^2 \; du \; dv$$

- Typically written as $E(\sigma) = \frac{1}{2} \int \int ||\nabla \sigma||^2 \; du \; dv$ or $E(\sigma) = \frac{1}{2} \int \int Tr(J\sigma^t J\sigma) \; du \; dv$, and is called the *Dirichlet Energy*.
- With conformal parameterization $\sqrt{EG - F^2} = E$, thus $A(\sigma) = E(\sigma)$.
- The minimizer of $E(\sigma)$ also satisfies $\nabla A(\sigma) = Hn = \Delta \sigma = 0$.
- Functions that satisfy[3] $\Delta \sigma = 0$ are called *Harmonic functions*.

[3] This is the famous Laplace equation.

Dirichlet Energy

- A parameterization σ of a surface maps points from $\mathbb{R}^2$ to points in $\mathbb{R}^3$.
- A measure of how rapidly σ varies:

$$E(\sigma) = \frac{1}{2}\int\int ||\frac{\partial \sigma}{\partial u}||^2 + ||\frac{\partial \sigma}{\partial v}||^2 \; du \; dv$$

- Typically written as $E(\sigma) = \frac{1}{2}\int\int ||\nabla\sigma||^2 \; du \; dv$ or $E(\sigma) = \frac{1}{2}\int\int Tr(J\sigma^t J\sigma) \; du \; dv$, and is called the *Dirichlet Energy*.
- With conformal parameterization $\sqrt{EG - F^2} = E$, thus $A(\sigma) = E(\sigma)$.
- The minimizer of $E(\sigma)$ also satisfies $\nabla A(\sigma) = Hn = \Delta\sigma = 0$.
- Functions that satisfy[3] $\Delta\sigma = 0$ are called *Harmonic functions*.

[3] This is the famous Laplace equation.

Dirichlet Energy

- A parameterization σ of a surface maps points from $\mathbb{R}^2$ to points in $\mathbb{R}^3$.
- A measure of how rapidly σ varies:

$$E(\sigma) = \frac{1}{2} \int \int ||\frac{\partial \sigma}{\partial u}||^2 + ||\frac{\partial \sigma}{\partial v}||^2 \; du \; dv$$

- Typically written as $E(\sigma) = \frac{1}{2} \int \int ||\nabla \sigma||^2 \; du \; dv$ or $E(\sigma) = \frac{1}{2} \int \int Tr(J\sigma^t J\sigma) \; du \; dv$, and is called the *Dirichlet Energy*.
- With conformal parameterization $\sqrt{EG - F^2} = E$, thus $A(\sigma) = E(\sigma)$.
- The minimizer of $E(\sigma)$ also satisfies $\nabla A(\sigma) = Hn = \Delta \sigma = 0$.
- Functions that satisfy[3] $\Delta \sigma = 0$ are called *Harmonic functions*.

[3] This is the famous Laplace equation.

Dirichlet Energy

- A parameterization σ of a surface maps points from $\mathbb{R}^2$ to points in $\mathbb{R}^3$.
- A measure of how rapidly σ varies:

$$E(\sigma) = \frac{1}{2} \int \int ||\frac{\partial \sigma}{\partial u}||^2 + ||\frac{\partial \sigma}{\partial v}||^2 \; du \; dv$$

- Typically written as $E(\sigma) = \frac{1}{2} \int \int ||\nabla \sigma||^2 \; du \; dv$ or $E(\sigma) = \frac{1}{2} \int \int Tr(J\sigma^t J\sigma) \; du \; dv$, and is called the *Dirichlet Energy*.
- With conformal parameterization $\sqrt{EG - F^2} = E$, thus $A(\sigma) = E(\sigma)$.
- The minimizer of $E(\sigma)$ also satisfies $\nabla A(\sigma) = Hn = \Delta \sigma = 0$.
- Functions that satisfy[3] $\Delta \sigma = 0$ are called *Harmonic functions*.

[3] This is the famous Laplace equation.

Dirichlet Energy

- A parameterization σ of a surface maps points from $\mathbb{R}^2$ to points in $\mathbb{R}^3$.
- A measure of how rapidly σ varies:

$$E(\sigma) = \frac{1}{2} \int \int ||\frac{\partial \sigma}{\partial u}||^2 + ||\frac{\partial \sigma}{\partial v}||^2 \; du \; dv$$

- Typically written as $E(\sigma) = \frac{1}{2} \int \int ||\nabla \sigma||^2 \; du \; dv$ or $E(\sigma) = \frac{1}{2} \int \int Tr(J\sigma^t J\sigma) \; du \; dv$, and is called the *Dirichlet Energy*.
- With conformal parameterization $\sqrt{EG - F^2} = E$, thus $A(\sigma) = E(\sigma)$.
- The minimizer of $E(\sigma)$ also satisfies $\nabla A(\sigma) = Hn = \Delta\sigma = 0$.
- Functions that satisfy[3] $\Delta\sigma = 0$ are called *Harmonic functions.*

[3] This is the famous Laplace equation.

Self-Adjoint operator

- Let $V := \{f : [0, 2\pi) \to \mathbb{R} \mid f \text{ is smooth and periodic}\}$, equipped with the standard L_2 inner product.
  - Note the following:

$$\int_0^{2\pi} \Delta f(x)\, g(x) dx = [\nabla f(x) g(x)]_0^{2\pi} - \int_0^{2\pi} \nabla f(x) \nabla g(x)\ dx$$

$$\text{Since } f, g \in V, \nabla f(0) = \nabla f(2\pi), g(0) = g(2\pi)$$

$$\begin{aligned} \int_0^{2\pi} \Delta f(x)\, g(x) dx &= - \int_0^{2\pi} \nabla f(x) \nabla g(x)\ dx \\ &= -[f(x) g(x)]_0^{2\pi} + \int_0^{2\pi} f(x) \Delta g(x)\ dx \\ &= \int_0^{2\pi} f(x) \Delta g(x)\ dx \end{aligned}$$

  - In terms of inner product, we get

$$\langle \Delta f, g \rangle_{L_2} = \langle f, \Delta g \rangle_{L_2} \Rightarrow \Delta^* = \Delta.$$

Self-Adjoint operator

- Let $V := \{f : [0, 2\pi) \to \mathbb{R} \mid f \text{ is smooth and periodic}\}$, equipped with the standard L_2 inner product.
- Note the following:

$$\int_0^{2\pi} \Delta f(x)\ g(x)dx = [\nabla f(x) g(x)]_0^{2\pi} - \int_0^{2\pi} \nabla f(x) \nabla g(x)\ dx$$

Since $f, g \in V, \nabla f(0) = \nabla f(2\pi), g(0) = g(2\pi)$

$$\begin{aligned}\int_0^{2\pi} \Delta f(x)\ g(x)dx &= -\int_0^{2\pi} \nabla f(x) \nabla g(x)\ dx \\ &= -[f(x)g(x)]_0^{2\pi} + \int_0^{2\pi} f(x) \Delta g(x)\ dx \\ &= \int_0^{2\pi} f(x) \Delta g(x)\ dx\end{aligned}$$

- In terms of inner product, we get

$$\langle \Delta f, g \rangle_{L_2} = \langle f, \Delta g \rangle_{L_2} \Rightarrow \Delta^* = \Delta.$$

Self-Adjoint operator

- Let $V := \{f : [0, 2\pi) \to \mathbb{R} \mid f \text{ is smooth and periodic}\}$, equipped with the standard L_2 inner product.
- Note the following:

$$\int_0^{2\pi} \Delta f(x)\ g(x)dx = [\nabla f(x)g(x)]_0^{2\pi} - \int_0^{2\pi} \nabla f(x)\nabla g(x)\ dx$$

Since $f, g \in V, \nabla f(0) = \nabla f(2\pi), g(0) = g(2\pi)$

$$\begin{aligned}\int_0^{2\pi} \Delta f(x)\ g(x)dx &= -\int_0^{2\pi} \nabla f(x)\nabla g(x)\ dx \\ &= -[f(x)g(x)]_0^{2\pi} + \int_0^{2\pi} f(x)\Delta g(x)\ dx \\ &= \int_0^{2\pi} f(x)\Delta g(x)\ dx\end{aligned}$$

- In terms of inner product, we get

$$\langle \Delta f, g\rangle_{L_2} = \langle f, \Delta g\rangle_{L_2} \Rightarrow \Delta^* = \Delta.$$

- Thus Δ can be diagonalized, and for the vector space V in the previous slide, the eigenfunctions of Δ are: $e_n(x) = \sin(nx)$ or $e_n(x) = \cos(nx), \forall n \in \mathbb{Z}$.
   - If complex valued functions are allowed, we have the familiar exponential Fourier series basis:

$$e_n(x) = \exp(jnx), \forall n \in \mathbb{Z}.$$

- Thus Δ can be diagonalized, and for the vector space V in the previous slide, the eigenfunctions of Δ are:
  $e_n(x) = \sin(nx)$ or $e_n(x) = \cos(nx), \forall n \in \mathbb{Z}$.
  - If complex valued functions are allowed, we have the familiar exponential Fourier series basis:

$$e_n(x) = \exp(jnx), \forall n \in \mathbb{Z}.$$

- Thus Δ can be diagonalized, and for the vector space V in the previous slide, the eigenfunctions of Δ are: $e_n(x) = \sin(nx)$ or $e_n(x) = \cos(nx), \forall n \in \mathbb{Z}$.
- If complex valued functions are allowed, we have the familiar exponential Fourier series basis:

$$e_n(x) = \exp(jnx), \forall n \in \mathbb{Z}.$$

Eigenfunctions of discrete LBO

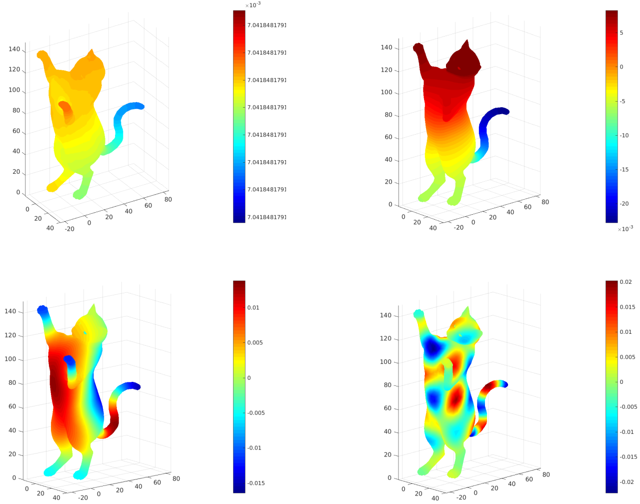


Figure : Can you interpret these eigenfunctions?

Deriving discrete LBO

- **Derivation 1.:** Minimizing Dirichlet energy[4].
- Let a surface be parameterized by the function f; the corresponding Dirichlet energy is

$$E_D(f) = \frac{1}{2} \int_\Omega ||\nabla f||^2 \, da$$

[4] Pinkall & Polthier, Computing Discrete Minimal Surfaces & their conjugates, Experimental Mathematics, vol. 2, no. 1, 1993

Deriving discrete LBO

- **Derivation 1.:** Minimizing Dirichlet energy[4].
- Let a surface be parameterized by the function f; the corresponding Dirichlet energy is

$$E_D(f) = \frac{1}{2} \int_\Omega ||\nabla f||^2 \, da$$

[4] Pinkall & Polthier, Computing Discrete Minimal Surfaces & their conjugates, Experimental Mathematics, vol. 2, no. 1, 1993

- Let us look at E_D triangle-wise.

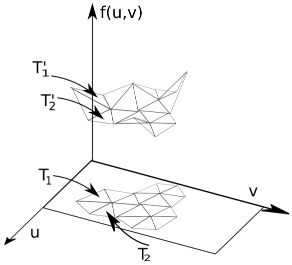


Figure : Triangles in local coordinates mapping to Triangles on the surface.

- For a discrete triangulated surface, we will assume that f is defined at vertices, while at all other points we will assume f to be linear.
  - Moreover, $E_D(f) = \sum_{i=1}^{m} E_D(f_i)$, where m is the total number of triangles, and f_i is a mapping between corresponding triangles i.

- Let us look at E_D triangle-wise.

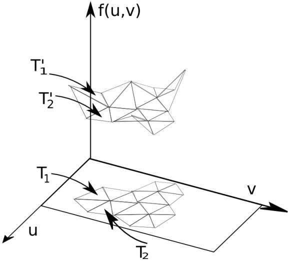


Figure : Triangles in local coordinates mapping to Triangles on the surface.

- For a discrete triangulated surface, we will assume that f is defined at vertices, while at all other points we will assume f to be linear.
  - Moreover, $E_D(f) = \sum_{i=1}^{m} E_D(f_i)$, where m is the total number of triangles, and f_i is a mapping between corresponding triangles i.

- Let us look at E_D triangle-wise.

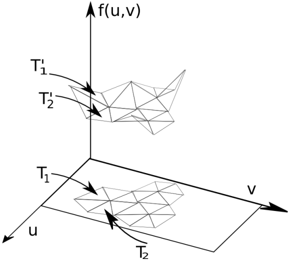


Figure : Triangles in local coordinates mapping to Triangles on the surface.

- For a discrete triangulated surface, we will assume that f is defined at vertices, while at all other points we will assume f to be linear.
  - Moreover, $E_D(f) = \sum_{i=1}^{m} E_D(f_i)$, where m is the total number of triangles, and f_i is a mapping between corresponding triangles i.

- The linear map $f_i : \Delta_1 \to \Delta_2$ is defined by $f(v) = a, f(w) = b$.

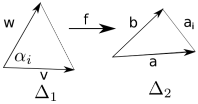


Figure : f_i between two corresponding triangles.

- The map f can also be written as $f = \psi \circ \phi^{-1}$, as shown in the following figure.

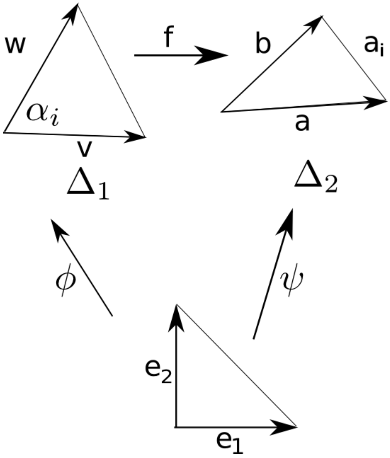


Figure : The map f via the standard triangle.

- Using Chain rule & Inverse Function theorem, $Jf = J\psi \circ J(\phi^{-1}) = J\psi \ (J\phi)^{-1}$.
  - Then,

$$\begin{aligned} Tr(Jf^t \ Jf) &= Tr((J\psi \ (J\phi)^{-1})^t J\psi \ (J\phi)^{-1}) \\ &= Tr(((J\phi)^{-1})^t J\psi^t J\psi \ (J\phi)^{-1}) \\ &= Tr(J\psi^t J\psi \ (J\phi)^{-1}((J\phi)^{-1})^t) \end{aligned}$$

  - Note that $J\psi = [a \quad b]$, thus $J\psi^t J\psi = \begin{bmatrix} \langle a, a \rangle & \langle a, b \rangle \\ \langle b, a \rangle & \langle b, b \rangle \end{bmatrix}$.
- Similarly, $J\phi = [v \quad w]$, thus

$(J\phi)^{-1}((J\phi)^{-1})^t = ((J\phi)^t J\phi)^{-1} = \begin{bmatrix} \langle v, v \rangle & \langle v, w \rangle \\ \langle w, v \rangle & \langle w, w \rangle \end{bmatrix}^{-1}$.

  - Let $d = det(J\phi)$ and $c = b - a$.

$$\begin{aligned} Tr(Jf^t \ Jf) = &\frac{1}{d} \left(\frac{\langle -w, v - w \rangle}{d} ||a||^2 + \frac{\langle w - v, -v \rangle}{d} ||b||^2 \right. \\ &\left. + \frac{\langle v, w \rangle}{d} ||c||^2 \right) \end{aligned}$$

- Using Chain rule & Inverse Function theorem, $Jf = J\psi \circ J(\phi^{-1}) = J\psi \, (J\phi)^{-1}$.
  - Then,

$$\begin{aligned} Tr(Jf^t \; Jf) &= Tr((J\psi \, (J\phi)^{-1})^t J\psi \, (J\phi)^{-1}) \\ &= Tr(((J\phi)^{-1})^t J\psi^t J\psi \, (J\phi)^{-1}) \\ &= Tr(J\psi^t J\psi \, (J\phi)^{-1}((J\phi)^{-1})^t) \end{aligned}$$

  - Note that $J\psi = [a \quad b]$, thus $J\psi^t J\psi = \begin{bmatrix} \langle a, a\rangle & \langle a, b\rangle \\ \langle b, a\rangle & \langle b, b\rangle \end{bmatrix}$.
- Similarly, $J\phi = [v \quad w]$, thus
  $(J\phi)^{-1}((J\phi)^{-1})^t = ((J\phi)^t J\phi)^{-1} = \begin{bmatrix} \langle v, v\rangle & \langle v, w\rangle \\ \langle w, v\rangle & \langle w, w\rangle \end{bmatrix}^{-1}$.
  - Let $d = det(J\phi)$ and $c = b - a$.

$$\begin{aligned} Tr(Jf^t \; Jf) =& \frac{1}{d}\left(\frac{\langle -w, v - w\rangle}{d}||a||^2 + \frac{\langle w - v, -v\rangle}{d}||b||^2 \right. \\ & \left. + \frac{\langle v, w\rangle}{d}||c||^2\right) \end{aligned}$$

- Using Chain rule & Inverse Function theorem,
  $Jf = J\psi \circ J(\phi^{-1}) = J\psi\ (J\phi)^{-1}$.
- Then,

$$\begin{aligned} Tr(Jf^t\ Jf) &= Tr((J\psi\ (J\phi)^{-1})^t J\psi\ (J\phi)^{-1}) \\ &= Tr(((J\phi)^{-1})^t J\psi^t J\psi\ (J\phi)^{-1}) \\ &= Tr(J\psi^t J\psi\ (J\phi)^{-1}((J\phi)^{-1})^t) \end{aligned}$$

- Note that $J\psi = [a \quad b]$, thus $J\psi^t J\psi = \begin{bmatrix} \langle a, a\rangle & \langle a, b\rangle \\ \langle b, a\rangle & \langle b, b\rangle \end{bmatrix}$.
- Similarly, $J\phi = [v \quad w]$, thus

$$(J\phi)^{-1}((J\phi)^{-1})^t = ((J\phi)^t J\phi)^{-1} = \begin{bmatrix} \langle v, v\rangle & \langle v, w\rangle \\ \langle w, v\rangle & \langle w, w\rangle \end{bmatrix}^{-1}.$$

- Let $d = det(J\phi)$ and $c = b - a$.

$$\begin{aligned} Tr(Jf^t\ Jf) =& \frac{1}{d}\left(\frac{\langle -w, v-w\rangle}{d}||a||^2 + \frac{\langle w-v, -v\rangle}{d}||b||^2 \right. \\ & \left. + \frac{\langle v, w\rangle}{d}||c||^2\right) \end{aligned}$$

- Using Chain rule & Inverse Function theorem, $Jf = J\psi \circ J(\phi^{-1}) = J\psi\ (J\phi)^{-1}$.
- Then,

$$\begin{aligned} Tr(Jf^t\ Jf) &= Tr((J\psi\ (J\phi)^{-1})^t J\psi\ (J\phi)^{-1}) \\ &= Tr(((J\phi)^{-1})^t J\psi^t J\psi\ (J\phi)^{-1}) \\ &= Tr(J\psi^t J\psi\ (J\phi)^{-1}((J\phi)^{-1})^t) \end{aligned}$$

- Note that $J\psi = [a \quad b]$, thus $J\psi^t J\psi = \begin{bmatrix} \langle a, a\rangle & \langle a, b\rangle \\ \langle b, a\rangle & \langle b, b\rangle \end{bmatrix}$.
  - Similarly, $J\phi = [v \quad w]$, thus

$$(J\phi)^{-1}((J\phi)^{-1})^t = ((J\phi)^t J\phi)^{-1} = \begin{bmatrix} \langle v, v\rangle & \langle v, w\rangle \\ \langle w, v\rangle & \langle w, w\rangle \end{bmatrix}^{-1}.$$

- Let $d = det(J\phi)$ and $c = b - a$.

$$Tr(Jf^t\ Jf) = \frac{1}{d}\left(\frac{\langle -w, v-w\rangle}{d}||a||^2 + \frac{\langle w-v, -v\rangle}{d}||b||^2 + \frac{\langle v, w\rangle}{d}||c||^2\right)$$

- Using Chain rule & Inverse Function theorem, $Jf = J\psi \circ J(\phi^{-1}) = J\psi\ (J\phi)^{-1}$.
  - Then,

$$\begin{aligned} Tr(Jf^t\ Jf) &= Tr((J\psi\ (J\phi)^{-1})^t J\psi\ (J\phi)^{-1}) \\ &= Tr(((J\phi)^{-1})^t J\psi^t J\psi\ (J\phi)^{-1}) \\ &= Tr(J\psi^t J\psi\ (J\phi)^{-1}((J\phi)^{-1})^t) \end{aligned}$$

  - Note that $J\psi = [a \quad b]$, thus $J\psi^t J\psi = \begin{bmatrix} \langle a, a\rangle & \langle a, b\rangle \\ \langle b, a\rangle & \langle b, b\rangle \end{bmatrix}$.
- Similarly, $J\phi = [v \quad w]$, thus

$$(J\phi)^{-1}((J\phi)^{-1})^t = ((J\phi)^t J\phi)^{-1} = \begin{bmatrix} \langle v, v\rangle & \langle v, w\rangle \\ \langle w, v\rangle & \langle w, w\rangle \end{bmatrix}^{-1}.$$

  - Let $d = det(J\phi)$ and $c = b - a$.

$$Tr(Jf^t\ Jf) = \frac{1}{d}\left(\frac{\langle -w, v-w\rangle}{d}||a||^2 + \frac{\langle w-v, -v\rangle}{d}||b||^2 + \frac{\langle v, w\rangle}{d}||c||^2\right)$$

- With $d = 2A$, and $d = ||v||\ ||w|| \sin\theta_1 = ||w||\ ||v-w|| \sin\theta_2 = ||w-v||\ ||v|| \sin\theta_3$, we get

$$Tr(Jf^t\ Jf) = \frac{1}{2A}\sum_{i=1}^{3} \cot\alpha_i ||a_i||^2$$

- The Dirichlet energy is given by

$$\begin{aligned} E_D(f_i) &= \frac{1}{2}\int_{\Delta_1} Tr(Jf_i^t Jf_i)\ da \\ &= \frac{1}{2} Tr(Jf_i^t Jf_i) \int_{\Delta_1} da \\ &= \frac{1}{4}\sum_{i=1}^{3} \cot\alpha_i ||a_i||^2 \end{aligned}$$

- The total Dirichlet energy is then,

$$E_D(f) = \sum_{i=1}^{m} E_D(f_i)$$

- With $d = 2A$, and $d = ||v||\ ||w|| \sin\theta_1 = ||w||\ ||v-w|| \sin\theta_2$ $= ||w-v||\ ||v|| \sin\theta_3$, we get

$$Tr(Jf^t\ Jf) = \frac{1}{2A}\sum_{i=1}^{3}\cot\alpha_i ||a_i||^2$$

- The Dirichlet energy is given by

$$\begin{aligned} E_D(f_i) &= \frac{1}{2}\int_{\Delta_1} Tr(Jf_i^t Jf_i)\ da \\ &= \frac{1}{2} Tr(Jf_i^t Jf_i)\int_{\Delta_1} da \\ &= \frac{1}{4}\sum_{i=1}^{3}\cot\alpha_i ||a_i||^2 \end{aligned}$$

- The total Dirichlet energy is then,

$$E_D(f) = \sum_{i=1}^{m} E_D(f_i)$$

- With $d = 2A$, and $d = ||v||\ ||w|| \sin\theta_1 = ||w||\ ||v - w|| \sin\theta_2$ $= ||w - v||\ ||v|| \sin\theta_3$, we get

$$Tr(Jf^t\ Jf) = \frac{1}{2A} \sum_{i=1}^{3} \cot\alpha_i ||a_i||^2$$

- The Dirichlet energy is given by

$$\begin{aligned} E_D(f_i) &= \frac{1}{2} \int_{\Delta_1} Tr(Jf_i^t Jf_i)\ da \\ &= \frac{1}{2} Tr(Jf_i^t Jf_i) \int_{\Delta_1}\ da \\ &= \frac{1}{4} \sum_{i=1}^{3} \cot\alpha_i ||a_i||^2 \end{aligned}$$

- The total Dirichlet energy is then,

$$E_D(f) = \sum_{i=1}^{m} E_D(f_i)$$

- To minimize Dirichlet energy, we differentiate w.r.t a vertex position and set it to zero:

$$\frac{\partial}{\partial p} E_D(f) = \frac{1}{2} \sum_{q \in N(p)} (\cot \alpha + \cot \beta)(p - q) = 0$$

- The minimizer can be interpreted as satisfying the (discrete) Laplace's equation: $Lf = (D - A)f = 0$, thus giving us the adjacency relation:

$$A_{ij} = \frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij})$$

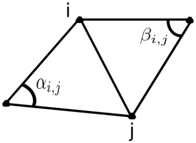


Figure : Weights for the Cotan Laplacian

- To minimize Dirichlet energy, we differentiate w.r.t a vertex position and set it to zero:

$$\frac{\partial}{\partial p} E_D(f) = \frac{1}{2} \sum_{q \in N(p)} (\cot \alpha + \cot \beta)(p - q) = 0$$

  - The minimizer can be interpreted as satisfying the (discrete) Laplace's equation: $Lf = (D - A)f = 0$, thus giving us the adjacency relation:

$$A_{ij} = \frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij})$$

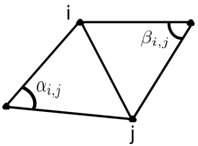


Figure : Weights for the Cotan Laplacian

Derivation 2: Weak formulation of the Poisson equation

- Compute solution to the Poisson's equation: $\Delta u = f$, i.e., given f, compute u.
  - The domain is a mesh, and the space of functions we work with is $V := \{u = \sum_j u_j B_j\}$, where B_j's are the linear functions with value 1 at vertex j, else 0, as shown below.

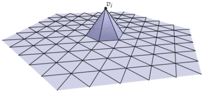


Figure : Piece-wise linear

  - Given f, can we compute $\hat{u} \in V$ as the best approximation to the actual solution u?
  - Idea 1: Given f, can we compute $\hat{u} \in V$ such that the laplacian of $\hat{u}$ best approximates the function f?
  - Let error $e = \Delta\hat{u} - f$. We want $e \in V^{\perp}$. $\Rightarrow \langle \Delta\hat{u} - f, B_i \rangle = 0, \forall i$.

Derivation 2: Weak formulation of the Poisson equation

- Compute solution to the Poisson's equation: $\Delta u = f$, i.e., given f, compute u.
  - The domain is a mesh, and the space of functions we work with is $V := \{u = \sum_j u_j B_j\}$, where B_j's are the linear functions with value 1 at vertex j, else 0, as shown below.

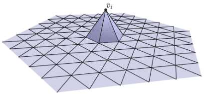


Figure : Piece-wise linear

  - Given f, can we compute $\hat{u} \in V$ as the best approximation to the actual solution u?
  - Idea 1: Given f, can we compute $\hat{u} \in V$ such that the laplacian of $\hat{u}$ best approximates the function f?
  - Let error $e = \Delta\hat{u} - f$. We want $e \in V^\perp$.
    $\Rightarrow \langle \Delta\hat{u} - f, B_i \rangle = 0, \forall i$.

Derivation 2: Weak formulation of the Poisson equation

- Compute solution to the Poisson's equation: $\Delta u = f$, i.e., given f, compute u.
- The domain is a mesh, and the space of functions we work with is $V := \{u = \sum_j u_j B_j\}$, where B_j's are the linear functions with value 1 at vertex j, else 0, as shown below.

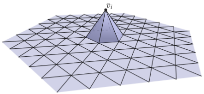


Figure : Piece-wise linear

- Given f, can we compute $\hat{u} \in V$ as the best approximation to the actual solution u?
- Idea 1: Given f, can we compute $\hat{u} \in V$ such that the laplacian of $\hat{u}$ best approximates the function f?
- Let error $e = \Delta \hat{u} - f$. We want $e \in V^{\perp}$.
  $\Rightarrow \langle \Delta \hat{u} - f, B_i \rangle = 0, \forall i.$

Derivation 2: Weak formulation of the Poisson equation

- Compute solution to the Poisson's equation: $\Delta u = f$, i.e., given f, compute u.
  - The domain is a mesh, and the space of functions we work with is $V := \{u = \sum_j u_j B_j\}$, where B_j's are the linear functions with value 1 at vertex j, else 0, as shown below.

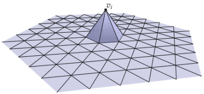


Figure : Piece-wise linear

  - Given f, can we compute $\hat{u} \in V$ as the best approximation to the actual solution u?
  - Idea 1: Given f, can we compute $\hat{u} \in V$ such that the laplacian of $\hat{u}$ best approximates the function f?
  - Let error $e = \Delta \hat{u} - f$. We want $e \in V^{\perp}$.
    $\Rightarrow \langle \Delta \hat{u} - f, B_i \rangle = 0, \forall i$.

Derivation 2: Weak formulation of the Poisson equation

- Compute solution to the Poisson's equation: $\Delta u = f$, i.e., given f, compute u.
  - The domain is a mesh, and the space of functions we work with is $V := \{u = \sum_j u_j B_j\}$, where B_j's are the linear functions with value 1 at vertex j, else 0, as shown below.

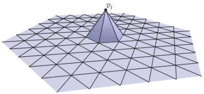


Figure : Piece-wise linear

  - Given f, can we compute $\hat{u} \in V$ as the best approximation to the actual solution u?
  - Idea 1: Given f, can we compute $\hat{u} \in V$ such that the laplacian of $\hat{u}$ best approximates the function f?
  - Let error $e = \Delta \hat{u} - f$. We want $e \in V^{\perp}$.
    $\Rightarrow \langle \Delta \hat{u} - f, B_i \rangle = 0, \forall i$.

- Inner product on the mesh Ω: $\langle f, g \rangle_\Omega = \int_\Omega f(x) g(x)\, da$.
- Solve: $\langle \Delta \hat{u}, B_i \rangle_\Omega = \langle f, B_i \rangle_\Omega \Rightarrow \sum_j u_j \langle \Delta B_j, B_i \rangle_\Omega = \langle f, B_i \rangle_\Omega$.
- Given that $\hat{u}$ (or B_j) $\in V$, how to define $\Delta \hat{u}$ (or ΔB_j).

- Inner product on the mesh Ω: $\langle f, g\rangle_\Omega = \int_\Omega f(x)g(x)\ da$.
- Solve: $\langle \Delta\hat{u}, B_i\rangle_\Omega = \langle f, B_i\rangle_\Omega \Rightarrow \sum_j u_j \langle \Delta B_j, B_i\rangle_\Omega = \langle f, B_i\rangle_\Omega$.
- Given that $\hat{u}$ (or B_j) $\in V$, how to define $\Delta\hat{u}$ (or ΔB_j).

- Inner product on the mesh Ω: $\langle f, g\rangle_\Omega = \int_\Omega f(x)g(x)\, da$.
- Solve: $\langle \Delta\hat{u}, B_i\rangle_\Omega = \langle f, B_i\rangle_\Omega \Rightarrow \sum_j u_j \langle \Delta B_j, B_i\rangle_\Omega = \langle f, B_i\rangle_\Omega$.
- Given that $\hat{u}$ (or B_j) $\in V$, how to define $\Delta\hat{u}$ (or ΔB_j).

- Integration by parts:

$$\int_{\Omega} f \Delta g da = \int_{\partial\Omega} f \langle \nabla g, n \rangle \; dl - \int_{\Omega} \langle \nabla f, \nabla g \rangle da,$$

where n is the outward unit normal to Ω on the boundary $\partial\Omega$.

- Note that ∇f and ∇g are vector fields on the mesh Ω, and the second term on the right hand side can be written as $\langle \nabla f, \nabla g \rangle_{\Omega}$.
- Idea 2: If Ω is a closed mesh, $\partial\Omega = \emptyset$, giving us

$$\langle f, \Delta g \rangle_{\Omega} = -\langle \nabla f, \nabla g \rangle_{\Omega}.$$

- Thus rather than solving $\sum_j u_j \langle \Delta B_j, B_i \rangle_{\Omega} = \langle f, B_i \rangle_{\Omega}$, we solve $-\sum_j u_j \langle \nabla B_j, \nabla B_i \rangle_{\Omega} = \langle f, B_i \rangle_{\Omega}$.
- Let $u = (u_1, \ldots, u_n)$, $\tilde{f} = (\langle f, B_1 \rangle_{\Omega}, \ldots, \langle f, B_n \rangle_{\Omega})$, and $L_{ij} = \langle \nabla B_i, \nabla B_j \rangle_{\Omega}$. Using these notations, solve $-Lu = \tilde{f}$.

- Integration by parts:

$$\int_{\Omega} f \Delta g da = \int_{\partial\Omega} f \langle \nabla g, n \rangle \; dl - \int_{\Omega} \langle \nabla f, \nabla g \rangle da,$$

where n is the outward unit normal to Ω on the boundary $\partial\Omega$.

- Note that ∇f and ∇g are vector fields on the mesh Ω, and the second term on the right hand side can be written as $\langle \nabla f, \nabla g \rangle_{\Omega}$.
- Idea 2: If Ω is a closed mesh, $\partial\Omega = \emptyset$, giving us

$$\langle f, \Delta g \rangle_{\Omega} = -\langle \nabla f, \nabla g \rangle_{\Omega}.$$

- Thus rather than solving $\sum_j u_j \langle \Delta B_j, B_i \rangle_{\Omega} = \langle f, B_i \rangle_{\Omega}$, we solve $-\sum_j u_j \langle \nabla B_j, \nabla B_i \rangle_{\Omega} = \langle f, B_i \rangle_{\Omega}$.
- Let $u = (u_1, \ldots, u_n)$, $\tilde{f} = (\langle f, B_1 \rangle_{\Omega}, \ldots, \langle f, B_n \rangle_{\Omega})$, and $L_{ij} = \langle \nabla B_i, \nabla B_j \rangle_{\Omega}$. Using these notations, solve $-Lu = \tilde{f}$.

- Integration by parts:

$$\int_\Omega f \Delta g da = \int_{\partial\Omega} f \langle \nabla g, n \rangle \; dl - \int_\Omega \langle \nabla f, \nabla g \rangle da,$$

where n is the outward unit normal to Ω on the boundary $\partial\Omega$.

- Note that ∇f and ∇g are vector fields on the mesh Ω, and the second term on the right hand side can be written as $\langle \nabla f, \nabla g \rangle_\Omega$.
- Idea 2: If Ω is a closed mesh, $\partial\Omega = \emptyset$, giving us

$$\langle f, \Delta g \rangle_\Omega = -\langle \nabla f, \nabla g \rangle_\Omega.$$

- Thus rather than solving $\sum_j u_j \langle \Delta B_j, B_i \rangle_\Omega = \langle f, B_i \rangle_\Omega$, we solve $-\sum_j u_j \langle \nabla B_j, \nabla B_i \rangle_\Omega = \langle f, B_i \rangle_\Omega$.
- Let $u = (u_1, \ldots, u_n)$, $\tilde{f} = (\langle f, B_1 \rangle_\Omega, \ldots, \langle f, B_n \rangle_\Omega)$, and $L_{ij} = \langle \nabla B_i, \nabla B_j \rangle_\Omega$. Using these notations, solve $-Lu = \tilde{f}$.

- Integration by parts:

$$\int_{\Omega} f \Delta g da = \int_{\partial\Omega} f \langle \nabla g, n \rangle \; dl - \int_{\Omega} \langle \nabla f, \nabla g \rangle da,$$

where n is the outward unit normal to Ω on the boundary $\partial\Omega$.

- Note that ∇f and ∇g are vector fields on the mesh Ω, and the second term on the right hand side can be written as $\langle \nabla f, \nabla g \rangle_{\Omega}$.
- Idea 2: If Ω is a closed mesh, $\partial\Omega = \emptyset$, giving us

$$\langle f, \Delta g \rangle_{\Omega} = -\langle \nabla f, \nabla g \rangle_{\Omega}.$$

- Thus rather than solving $\sum_j u_j \langle \Delta B_j, B_i \rangle_{\Omega} = \langle f, B_i \rangle_{\Omega}$, we solve $-\sum_j u_j \langle \nabla B_j, \nabla B_i \rangle_{\Omega} = \langle f, B_i \rangle_{\Omega}$.
- Let $u = (u_1, \ldots, u_n)$, $\tilde{f} = (\langle f, B_1 \rangle_{\Omega}, \ldots, \langle f, B_n \rangle_{\Omega})$, and $L_{ij} = \langle \nabla B_i, \nabla B_j \rangle_{\Omega}$. Using these notations, solve $-Lu = \tilde{f}$.

- Integration by parts:

$$\int_\Omega f \Delta g da = \int_{\partial\Omega} f \langle \nabla g, n \rangle \; dl - \int_\Omega \langle \nabla f, \nabla g \rangle da,$$

where n is the outward unit normal to Ω on the boundary $\partial\Omega$.

- Note that ∇f and ∇g are vector fields on the mesh Ω, and the second term on the right hand side can be written as $\langle \nabla f, \nabla g \rangle_\Omega$.
- Idea 2: If Ω is a closed mesh, $\partial\Omega = \emptyset$, giving us

$$\langle f, \Delta g \rangle_\Omega = -\langle \nabla f, \nabla g \rangle_\Omega.$$

- Thus rather than solving $\sum_j u_j \langle \Delta B_j, B_i \rangle_\Omega = \langle f, B_i \rangle_\Omega$, we solve $-\sum_j u_j \langle \nabla B_j, \nabla B_i \rangle_\Omega = \langle f, B_i \rangle_\Omega$.
- Let $u = (u_1, \ldots, u_n)$, $\tilde{f} = (\langle f, B_1 \rangle_\Omega, \ldots, \langle f, B_n \rangle_\Omega)$, and $L_{ij} = \langle \nabla B_i, \nabla B_j \rangle_\Omega$. Using these notations, solve $-Lu = \tilde{f}$.

- The discrete Laplacian of the function u on the mesh at vertex i is $(Lu)_i$.
   - $\langle \nabla B_i, \nabla B_i \rangle_\Omega = \sum_k \langle \nabla B_i, \nabla B_i \rangle_{\Delta_k}$.
   - $\langle \nabla B_i, \nabla B_i \rangle_{\Delta_k} = \frac{1}{2}(\cot\alpha + \cot\beta)$.
   - $\langle \nabla B_i, \nabla B_j \rangle_\Omega = \sum_k \langle \nabla B_i, \nabla B_j \rangle_{\Delta_k}$
   - $\langle \nabla B_i, \nabla B_j \rangle_{\Delta_k} = -\frac{1}{2}\cot\theta$.

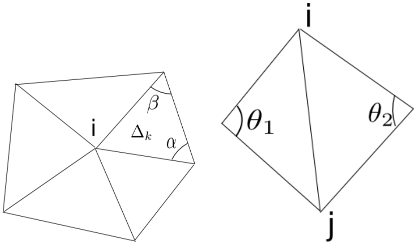


Figure : Laplacian at a point in the mesh

- The discrete Laplacian of the function u on the mesh at vertex i is $(Lu)_i$.

- $\langle \nabla B_i, \nabla B_i \rangle_\Omega = \sum_k \langle \nabla B_i, \nabla B_i \rangle_{\Delta_k}$.
- $\langle \nabla B_i, \nabla B_i \rangle_{\Delta_k} = \frac{1}{2}(\cot\alpha + \cot\beta)$.
- $\langle \nabla B_i, \nabla B_j \rangle_\Omega = \sum_k \langle \nabla B_i, \nabla B_j \rangle_{\Delta_k}$
- $\langle \nabla B_i, \nabla B_j \rangle_{\Delta_k} = -\frac{1}{2}\cot\theta$.

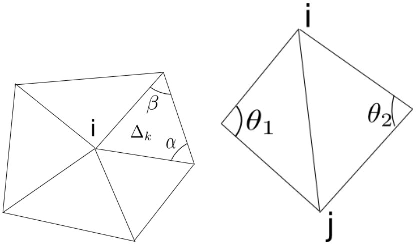


Figure : Laplacian at a point in the mesh

- The discrete Laplacian of the function u on the mesh at vertex i is $(Lu)_i$.

  - $\langle \nabla B_i, \nabla B_i \rangle_\Omega = \sum_k \langle \nabla B_i, \nabla B_i \rangle_{\Delta_k}$.
  - $\langle \nabla B_i, \nabla B_i \rangle_{\Delta_k} = \frac{1}{2}(\cot\alpha + \cot\beta)$.
  - $\langle \nabla B_i, \nabla B_j \rangle_\Omega = \sum_k \langle \nabla B_i, \nabla B_j \rangle_{\Delta_k}$
  - $\langle \nabla B_i, \nabla B_j \rangle_{\Delta_k} = -\frac{1}{2}\cot\theta$.

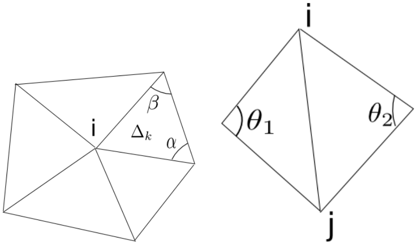


Figure : Laplacian at a point in the mesh

- The discrete Laplacian of the function u on the mesh at vertex i is $(Lu)_i$.

- $\langle \nabla B_i, \nabla B_i \rangle_\Omega = \sum_k \langle \nabla B_i, \nabla B_i \rangle_{\Delta_k}$.
- $\langle \nabla B_i, \nabla B_i \rangle_{\Delta_k} = \frac{1}{2}(\cot\alpha + \cot\beta)$.
- $\langle \nabla B_i, \nabla B_j \rangle_\Omega = \sum_k \langle \nabla B_i, \nabla B_j \rangle_{\Delta_k}$
- $\langle \nabla B_i, \nabla B_j \rangle_{\Delta_k} = -\frac{1}{2}\cot\theta$.

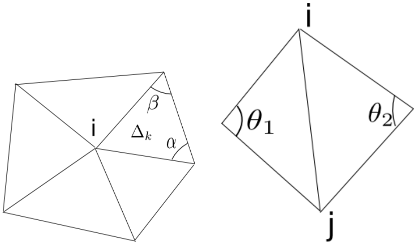


Figure : Laplacian at a point in the mesh

- The discrete Laplacian of the function u on the mesh at vertex i is $(Lu)_i$.
  - $\langle \nabla B_i, \nabla B_i \rangle_\Omega = \sum_k \langle \nabla B_i, \nabla B_i \rangle_{\Delta_k}$.
  - $\langle \nabla B_i, \nabla B_i \rangle_{\Delta_k} = \frac{1}{2}(\cot\alpha + \cot\beta)$.
  - $\langle \nabla B_i, \nabla B_j \rangle_\Omega = \sum_k \langle \nabla B_i, \nabla B_j \rangle_{\Delta_k}$
  - $\langle \nabla B_i, \nabla B_j \rangle_{\Delta_k} = -\frac{1}{2}\cot\theta$.

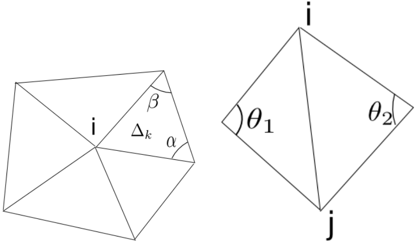


Figure : Laplacian at a point in the mesh

- Thus $\langle \nabla B_i, \nabla B_i \rangle_\Omega = \frac{1}{2}\sum_{\Delta_i}(\cot\alpha_i + \cot\beta_i)$ and $\langle \nabla B_i, \nabla B_j \rangle_\Omega = -\frac{1}{2}(\cot\theta_1 + \cot\theta_2)$.
- The Discrete LBO is

$$
\begin{aligned}
L_{ii} =& \frac{1}{2}\sum_{i\sim k}(\cot\alpha_k + \cot\beta_k) \\
L_{ij} =& -\frac{1}{2}(\cot\alpha_j + \cot\beta_j), \quad \text{if } i \sim j. \\
L_{ij} =& 0, \quad \text{otherwise.}
\end{aligned}
$$

  - Note that in this case $L = D - A$, you may want to use $\hat{L} = -L$.

- Thus $\langle \nabla B_i, \nabla B_i \rangle_\Omega = \frac{1}{2}\sum_{\Delta_i}(\cot\alpha_i + \cot\beta_i)$ and $\langle \nabla B_i, \nabla B_j \rangle_\Omega = -\frac{1}{2}(\cot\theta_1 + \cot\theta_2)$.
- The Discrete LBO is

$$
\begin{aligned}
L_{ii} =& \frac{1}{2}\sum_{i\sim k}(\cot\alpha_k + \cot\beta_k)\\
L_{ij} =& -\frac{1}{2}(\cot\alpha_j + \cot\beta_j), \quad \text{if } i \sim j.\\
L_{ij} =& 0, \quad \text{otherwise.}
\end{aligned}
$$

- Note that in this case $L = D - A$, you may want to use $\hat{L} = -L$.

- Thus $\langle \nabla B_i, \nabla B_i \rangle_\Omega = \frac{1}{2} \sum_{\Delta_i} (\cot \alpha_i + \cot \beta_i)$ and $\langle \nabla B_i, \nabla B_j \rangle_\Omega = -\frac{1}{2} (\cot \theta_1 + \cot \theta_2)$.
- The Discrete LBO is

$$
\begin{aligned}
L_{ii} =& \frac{1}{2} \sum_{i \sim k} (\cot \alpha_k + \cot \beta_k) \\
L_{ij} =& -\frac{1}{2} (\cot \alpha_j + \cot \beta_j), \quad \text{if } i \sim j. \\
L_{ij} =& 0, \quad \text{otherwise.}
\end{aligned}
$$

- Note that in this case $L = D - A$, you may want to use $\hat{L} = -L$.

Cotan Laplacian[5]

- Using the definition of LBO: $\nabla = div \circ grad$:

$$\int_{A_i} \Delta f(x)\ da = \int_{A_i} div \nabla f(x)\ da = \int_{\partial A_i} \nabla f(x) \cdot n(x)\ dl,$$

where A_i refers to a local integration (averaging) area, typically the Voronoi cell of a vertex.

- This reduces to

$$\int_{A_i} \Delta f(x)\ da = \frac{1}{2} \sum_{j \in N(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(f_j - f_i)$$

- Thus the LBO on function f is:

$$\Delta f(v_i) = \frac{1}{2A_i} \sum_{j \in N(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(f_j - f_i)$$

[5] Meyer et.al., Discrete Differential Geometry Operators for Triangulated 2-Manifolds, Vis. & Math - 3, 2003

Cotan Laplacian[5]

- Using the definition of LBO: $\nabla = div \circ grad$:

$$\int_{A_i} \Delta f(x)\ da = \int_{A_i} div \nabla f(x)\ da = \int_{\partial A_i} \nabla f(x) \cdot n(x)\ dl,$$

where A_i refers to a local integration (averaging) area, typically the Voronoi cell of a vertex.

- This reduces to

$$\int_{A_i} \Delta f(x)\ da = \frac{1}{2} \sum_{j \in N(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(f_j - f_i)$$

- Thus the LBO on function f is:

$$\Delta f(v_i) = \frac{1}{2A_i} \sum_{j \in N(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(f_j - f_i)$$

[5] Meyer et.al., Discrete Differential Geometry Operators for Triangulated 2-Manifolds, Vis. & Math - 3, 2003

Cotan Laplacian[5]

- Using the definition of LBO: $\nabla = div \circ grad$:

$$\int_{A_i} \Delta f(x)\ da = \int_{A_i} div \nabla f(x)\ da = \int_{\partial A_i} \nabla f(x) \cdot n(x)\ dl,$$

where A_i refers to a local integration (averaging) area, typically the Voronoi cell of a vertex.

- This reduces to

$$\int_{A_i} \Delta f(x)\ da = \frac{1}{2} \sum_{j \in N(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(f_j - f_i)$$

- Thus the LBO on function f is:

$$\Delta f(v_i) = \frac{1}{2A_i} \sum_{j \in N(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(f_j - f_i)$$

[5] Meyer et.al., Discrete Differential Geometry Operators for Triangulated 2-Manifolds, Vis. & Math - 3, 2003

- Solution to Poisson eqn: $\Delta u = f$. What if $f \in V$?
  - $\langle f, B_i \rangle_\Omega = \langle \sum_j f_j B_j, B_i \rangle_\Omega = \sum_j f_j \langle B_j, B_i \rangle_\Omega$.
  - Let $A_{ij} = \int_\Omega B_i(x) B_j(x)\, da$. Then $\tilde{f} = A(f_1, \ldots, f_n)^t = A\bar{f}$.
  - The matrix A is called the *Mass matrix*, while the discrete Poisson eqn becomes $Lu = A\bar{f}$.
  - The matrix $A^{-1}L$ is called the *Area normalized Cotan Laplacian*.
  - One can derive the following:

$$A_{ii} = \frac{\text{Area of one-ring of vertex } i}{6}$$

$$A_{ij} = \frac{\text{Area of common triangles}}{12}$$

- Solution to Poisson eqn: $\Delta u = f$. What if $f \in V$?
  - $\langle f, B_i \rangle_\Omega = \langle \sum_j f_j B_j, B_i \rangle_\Omega = \sum_j f_j \langle B_j, B_i \rangle_\Omega$.
  - Let $A_{ij} = \int_\Omega B_i(x) B_j(x)\ da$. Then $\tilde{f} = A(f_1, \ldots, f_n)^t = A\bar{f}$.
  - The matrix A is called the *Mass matrix*, while the discrete Poisson eqn becomes $Lu = A\bar{f}$.
  - The matrix $A^{-1}L$ is called the *Area normalized Cotan Laplacian*.
  - One can derive the following:

$$A_{ii} = \frac{\text{Area of one-ring of vertex } i}{6}$$
$$A_{ij} = \frac{\text{Area of common triangles}}{12}$$

- Solution to Poisson eqn: $\Delta u = f$. What if $f \in V$?
  - $\langle f, B_i \rangle_\Omega = \langle \sum_j f_j B_j, B_i \rangle_\Omega = \sum_j f_j \langle B_j, B_i \rangle_\Omega$.
  - Let $A_{ij} = \int_\Omega B_i(x) B_j(x)\ da$. Then $\tilde{f} = A(f_1, \ldots, f_n)^t = A\bar{f}$.
  - The matrix A is called the *Mass matrix*, while the discrete Poisson eqn becomes $Lu = A\bar{f}$.
  - The matrix $A^{-1}L$ is called the *Area normalized Cotan Laplacian*.
  - One can derive the following:

$$A_{ii} = \frac{\text{Area of one-ring of vertex } i}{6}$$

$$A_{ij} = \frac{\text{Area of common triangles}}{12}$$

- Solution to Poisson eqn: $\Delta u = f$. What if $f \in V$?
  - $\langle f, B_i \rangle_\Omega = \langle \sum_j f_j B_j, B_i \rangle_\Omega = \sum_j f_j \langle B_j, B_i \rangle_\Omega$.
  - Let $A_{ij} = \int_\Omega B_i(x) B_j(x)\ da$. Then $\tilde{f} = A(f_1, \ldots, f_n)^t = A\bar{f}$.
  - The matrix A is called the *Mass matrix*, while the discrete Poisson eqn becomes $Lu = A\bar{f}$.
  - The matrix $A^{-1}L$ is called the *Area normalized Cotan Laplacian*.
  - One can derive the following:

$$A_{ii} = \frac{\text{Area of one-ring of vertex } i}{6}$$

$$A_{ij} = \frac{\text{Area of common triangles}}{12}$$

- Solution to Poisson eqn: $\Delta u = f$. What if $f \in V$?
  - $\langle f, B_i \rangle_\Omega = \langle \sum_j f_j B_j, B_i \rangle_\Omega = \sum_j f_j \langle B_j, B_i \rangle_\Omega$.
  - Let $A_{ij} = \int_\Omega B_i(x) B_j(x)\ da$. Then $\tilde{f} = A(f_1, \ldots, f_n)^t = A\bar{f}$.
  - The matrix A is called the *Mass matrix*, while the discrete Poisson eqn becomes $Lu = A\bar{f}$.
  - The matrix $A^{-1}L$ is called the *Area normalized Cotan Laplacian*.
  - One can derive the following:

$$A_{ii} = \frac{\text{Area of one-ring of vertex } i}{6}$$
$$A_{ij} = \frac{\text{Area of common triangles}}{12}$$

- Solution to Poisson eqn: $\Delta u = f$. What if $f \in V$?
  - $\langle f, B_i \rangle_\Omega = \langle \sum_j f_j B_j, B_i \rangle_\Omega = \sum_j f_j \langle B_j, B_i \rangle_\Omega$.
  - Let $A_{ij} = \int_\Omega B_i(x) B_j(x) \; da$. Then $\tilde{f} = A(f_1, \ldots, f_n)^t = A\bar{f}$.
  - The matrix A is called the *Mass matrix*, while the discrete Poisson eqn becomes $Lu = A\bar{f}$.
  - The matrix $A^{-1}L$ is called the *Area normalized Cotan Laplacian*.
  - One can derive the following:

$$A_{ii} = \frac{\text{Area of one-ring of vertex } i}{6}$$
$$A_{ij} = \frac{\text{Area of common triangles}}{12}$$

Application 1: Heat Equation

- $\frac{\partial u}{\partial t} = \Delta u.$

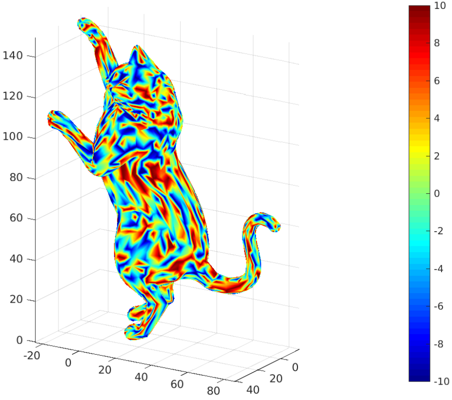


Figure : Heat diffusion on a cat.

Application 2: Compression

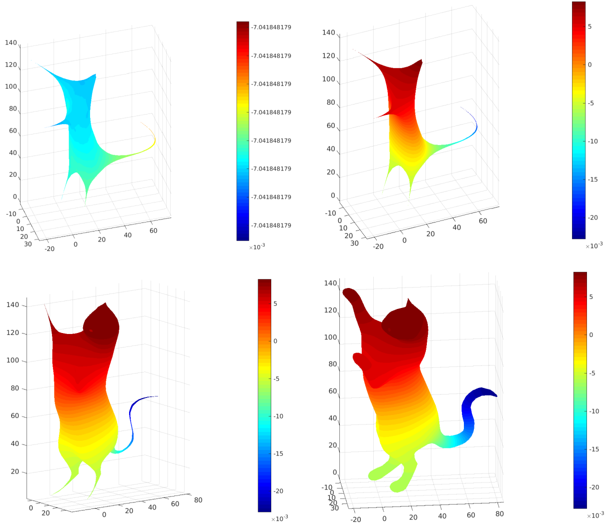


Figure : Mesh recovered using (top-left) 1, (top-right) 10, (bottom-left) 50 and (bottom-right) 200 coefficients, out of a total 3400 coefficients.