Discrete differential geometry

Xiao-Ming Fu

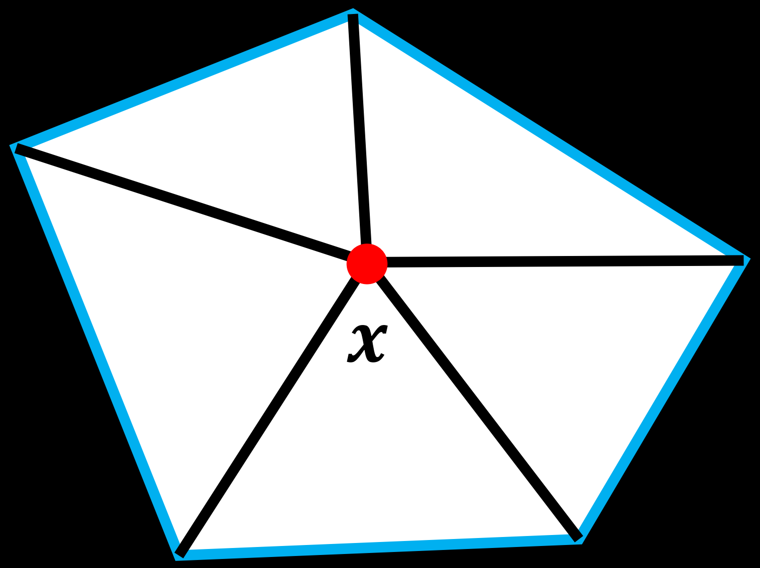
Goal

- Compute approximations of the differential properties of this underlying surface directly from the mesh data.

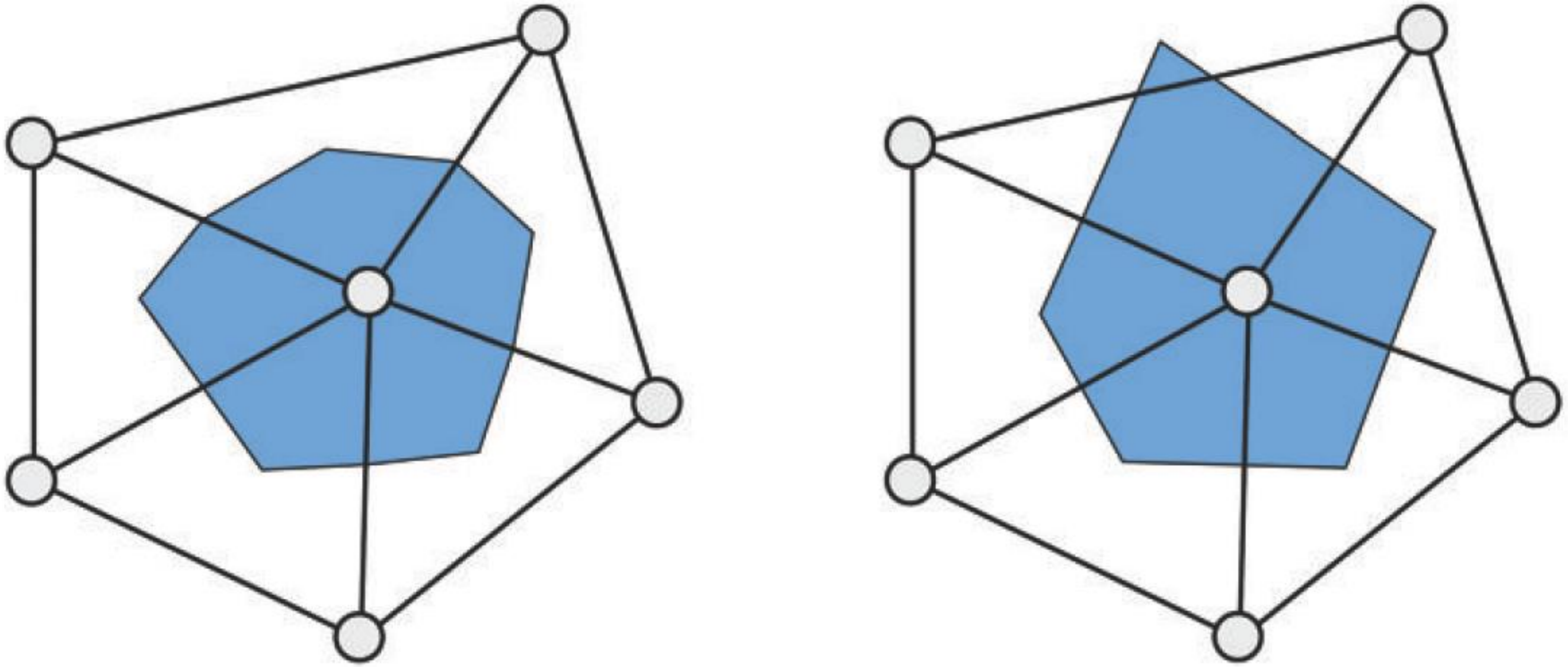
- Local Averaging Region
- Normal Vectors
- Gradients
- Laplace-Beltrami Operator
- Discrete Curvature

Local Averaging Region

- General idea: spatial averages over a local neighborhood $\Omega(\boldsymbol{x})$ of a point $\boldsymbol{x}$.
- $\boldsymbol{x}$: one mesh vertex
- $\Omega(\boldsymbol{x})$: n-ring neighborhoods of mesh vertex or local geodesic balls.
- The size of the $\Omega(\boldsymbol{x})$: stability and accuracy
  - Large size: smooth
  - Small size: accurate for clean mesh data



Local Averaging Region



Barycentric cell | Voronoi cell | Mixed Voronoi cell

triangle barycenters
edge midpoints

triangle barycenters
→ triangle circumcenter

circumcenter for obtuse
triangles → edge midpoints

Implementation thinking

- How to compute the area of local average region? For example, barycentric cell.
- One simple idea: for each vertex, compute the area directly.
- Any improvement?
- How about Voronoi cell and mixed Voronoi cell?

Normal Vectors

- Normal vectors for individual triangles are well-defined.
- Vertex normal: spatial averages of normal vectors in a local one-ring neighborhood.

$$\boldsymbol{n}(v) = \frac{\sum_{T\in\Omega(v)} \alpha_T \boldsymbol{n}(T)}{\left\|\sum_{T\in\Omega(v)} \alpha_T \boldsymbol{n}(T)\right\|_2}$$

1. constant weights: $\alpha_T = 1$
2. triangle area: $\alpha_T = \mathrm{area}(T)$
3. incident triangle angles: $\alpha_T = \theta(T)$

Implementation thinking

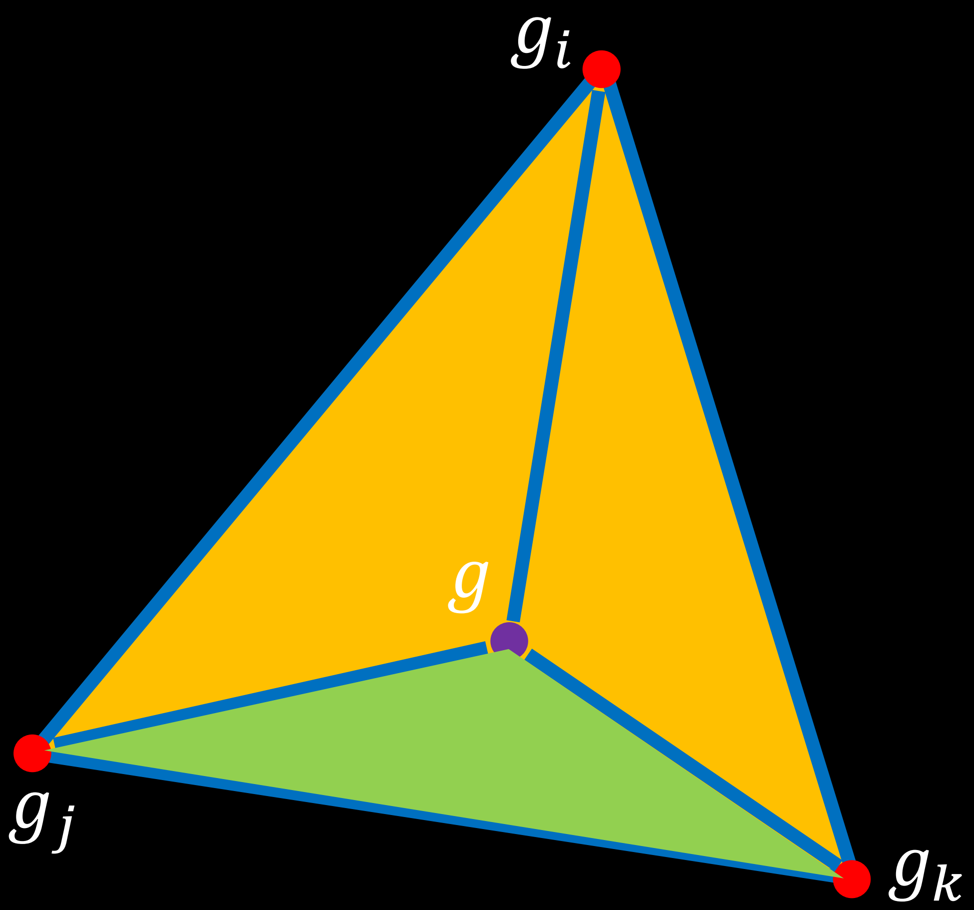
- How to compute the normal on triangles or vertices?

Barycentric coordinate

$$g = \alpha g_i + \beta g_j + \gamma g_k$$
$$\alpha + \beta + \gamma = 1,$$
$$\alpha, \beta, \gamma \geq 0\,.$$

$$\alpha = \frac{s_i}{s_i + s_j + s_k}$$

s_i: area of the green triangle



Gradients

- Given the function value on vertices, compute the gradient on each triangle.
- A piecewise linear function

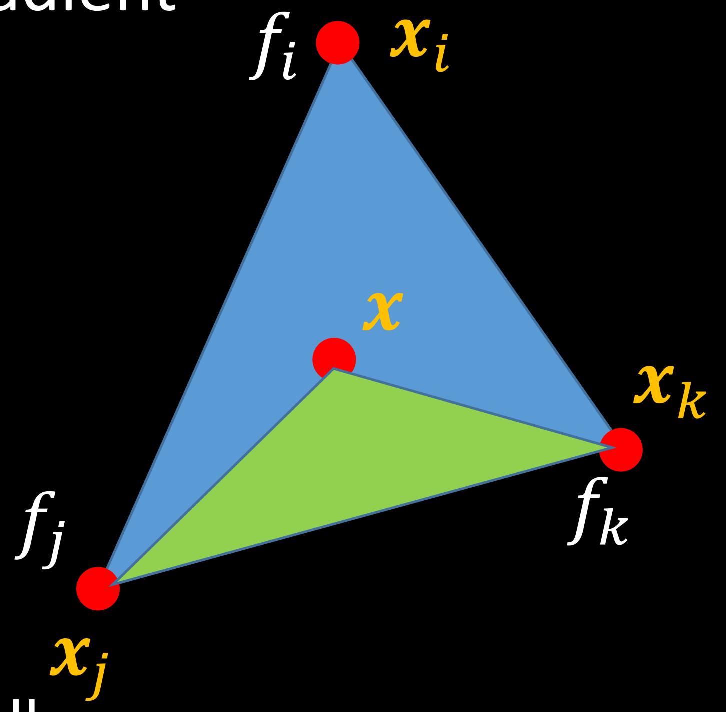
$$f(\boldsymbol{x}) = \alpha f_i + \beta f_j + \gamma f_k$$

- Gradient:

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = f_i \nabla_{\boldsymbol{x}} \alpha + f_j \nabla_{\boldsymbol{x}} \beta + f_k \nabla_{\boldsymbol{x}} \gamma$$

- Because

$$\alpha = \frac{A_i}{A_T} = \frac{\left((\boldsymbol{x} - \boldsymbol{x}_j) \cdot \frac{(\boldsymbol{x}_k - \boldsymbol{x}_j)^{\perp}}{\|\boldsymbol{x}_k - \boldsymbol{x}_j\|_2} \right) \|\boldsymbol{x}_k - \boldsymbol{x}_j\|_2}{2A_T}$$
$$= (\boldsymbol{x} - \boldsymbol{x}_j) \cdot (\boldsymbol{x}_k - \boldsymbol{x}_j)^{\perp} / 2A_T$$



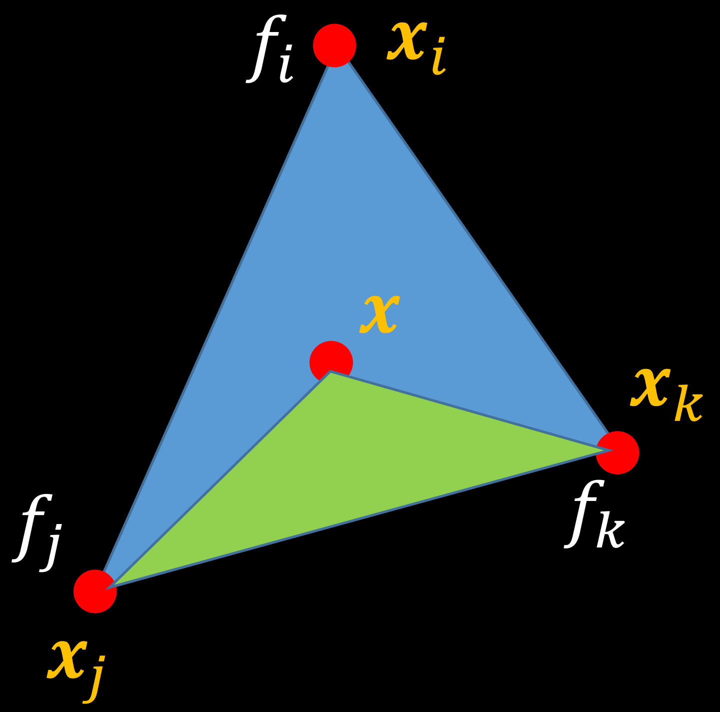
Gradients

- Then

$$\nabla_{\boldsymbol{x}}\alpha = \frac{\left(\boldsymbol{x}_k - \boldsymbol{x}_j\right)^{\perp}}{2A_T}$$

$$\nabla_{\boldsymbol{x}}\beta = \frac{\left(\boldsymbol{x}_i - \boldsymbol{x}_k\right)^{\perp}}{2A_T}$$

$$\nabla_{\boldsymbol{x}}\gamma = \frac{\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)^{\perp}}{2A_T}$$



=>

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = f_i \frac{\left(\boldsymbol{x}_k - \boldsymbol{x}_j\right)^{\perp}}{2A_T} + f_j \frac{\left(\boldsymbol{x}_i - \boldsymbol{x}_k\right)^{\perp}}{2A_T} + f_k \frac{\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)^{\perp}}{2A_T}$$

Gradients

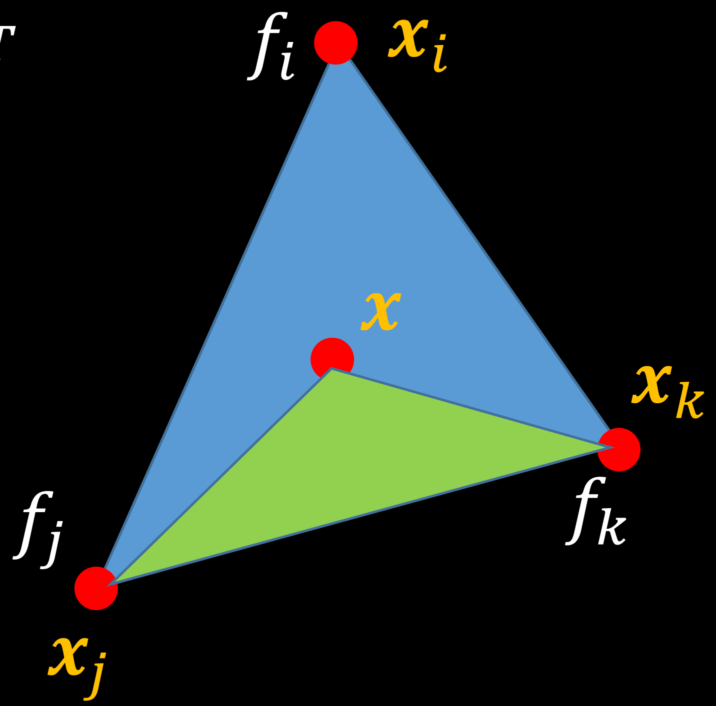
- Because:

$$(\boldsymbol{x}_k - \boldsymbol{x}_j)^\perp + (\boldsymbol{x}_i - \boldsymbol{x}_k)^\perp + (\boldsymbol{x}_j - \boldsymbol{x}_i)^\perp = 0$$

=>

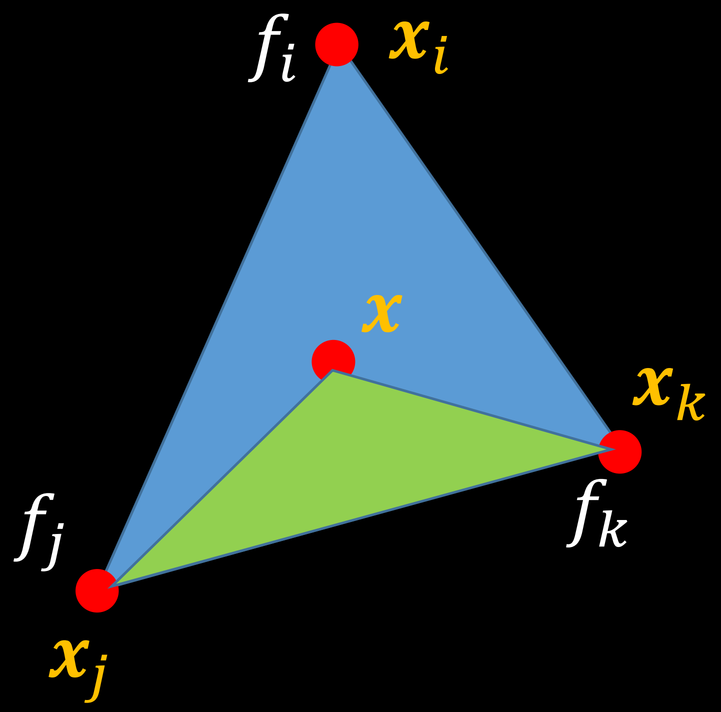
$$\nabla_x f(\boldsymbol{x}) = (f_j - f_i)\frac{(\boldsymbol{x}_i - \boldsymbol{x}_k)^\perp}{2A_T} + (f_k - f_i)\frac{(\boldsymbol{x}_j - \boldsymbol{x}_i)^\perp}{2A_T}$$

- Consistent with the formula in the book.



Gradient

- Constant on each facet.
- Different in different facets
  - the signal is C^0
- No definition on vertices.
- If the signal is the positions of the vertices, what does the gradient mean?

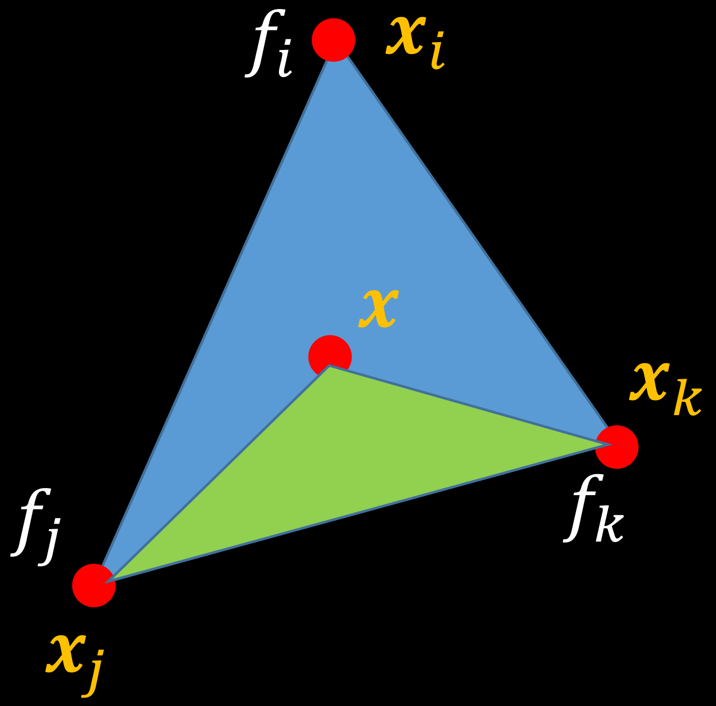


Implementation thinking

- Simple question:

  How to compute $(\boldsymbol{x}_k - \boldsymbol{x}_j)^{\perp}$ in 3D?

- How about the gradient in tetrahedral mesh?



Laplace-Beltrami Operator
Paper: Discrete Laplace operators: No free lunch

- $\Delta = -\mathrm{div}\,\mathrm{grad}$ on a smooth surface S
  - (NULL): $\Delta f = 0$ whenever f is constant.
  - (SYM) Symmetry: $(\Delta f, g)_{L^2} = (f, \Delta g)_{L^2}$
  - (LOC) Local support: for any pair $p \neq q$ of points, $\Delta f(p)$ is independent of $f(q)$.
  - (LIN) Linear precision: $\Delta f = 0$ whenever $S \in R^2$ and f is linear.
  - (MAX) Maximum principle: harmonic functions have no local maxima at interior points.
  - (PSD) Positive semi-definiteness: the Dirichlet energy $E_D(f) = \int_S \|\mathrm{grad}\, f\|_2^2\, dA = (\Delta f, f)_{L^2}$ is non-negative.

Discrete Laplace-Beltrami Operator
Paper: Discrete Laplace operators: No free lunch

- Discrete Laplace operators on triangular surface meshes span the entire spectrum of geometry processing applications:
  - mesh filtering, parameterization, pose transfer, segmentation, reconstruction, re-meshing, compression, simulation, and interpolation via barycentric coordinates.

- Constant gradient on facet $\rightarrow$ zero Laplace value on facet
- Exists on the vertex
- A discrete Laplace operator on vertex-based functions:

$$(\mathrm{L}f)_i = \sum_{j \in \Omega(i)} \omega_{ij} \left(f_j - f_i\right)$$

Desired Properties for Discrete Laplace-Beltrami Operator

- Require a discrete Laplacian having properties corresponding to (some subset of) the properties of the continuous Laplace operator:
  - NULL
    - $\Delta f = 0$ whenever f is constant.
  - SYM (SYMMETRY)
    - Condition: $\omega_{ij} = \omega_{ji}$
    - Real symmetric matrices exhibit real eigenvalues and orthogonal eigenvectors.
  - LOC (LOCALITY)
    - Condition: Weights are associated to mesh edges, $\omega_{ij} = 0$ if vertex i and j do not share an edge.
    - Smooth Laplacians govern diffusion processes via $u_t = -\Delta f$.
  - LIN (LINEAR PRECISION)
    - $(\mathrm{L}f)_i = 0$ for all interior vertices when the positions of vertices are in the plane.
    - Condition: $0 = (L\boldsymbol{x})_i = \sum_j \omega_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)$
    - Applications: de-noising, parameterizations, plate bending energies.

Desired Properties for Discrete Laplace-Beltrami Operator

- POS (POSITIVE WEIGHTS)
  - Condition: $\omega_{ij} > 0$ whenever $i \neq j$.
  - A sufficient condition for a discrete maximum principle.
  - In diffusion problems, this property assures that flow travels from regions of higher to regions of lower potential.
  - Establishes a connection to barycentric coordinates.
  - Tutte's embedding theorem: LOCALITY + LINEAR PRECISION + POSITIVE WEIGHTS.
- PSD (POSITIVE SEMI-DEFINITENESS)
  - Condition: L is symmetric positive semi-definite.
  - Discrete Dirichlet energy $E_D(f) = \sum_{i,j} \omega_{ij} \left(f_i - f_j\right)^2$.
  - SYMMETRY + POSITIVE WEIGHTS $\rightarrow$ POSITIVE SEMI-DEFINITENESS
  - POSITIVE SEMI-DEFINITENESS $\nrightarrow$ POSITIVE WEIGHTS

Uniform Laplacian

- $\omega_{ij} = 1$ or $\omega_{ij} = \frac{1}{N_i}$

$(\mathrm{L}f)_i = \sum_{j \in \Omega(i)} (f_j - f_i)$ or $(\mathrm{L}f)_i = \frac{1}{N_i} \sum_{j \in \Omega(i)} (f_j - f_i)$

- Violate property of LINEAR PRECISION

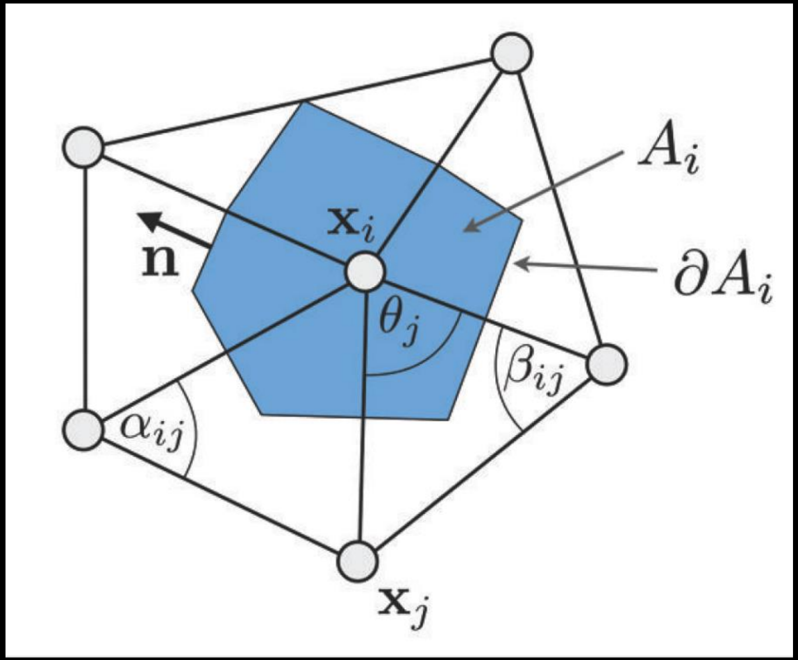
- The definition only depends on the connectivity of the mesh.
- The uniform Laplacian does not adapt at all to the spatial distribution of vertices.

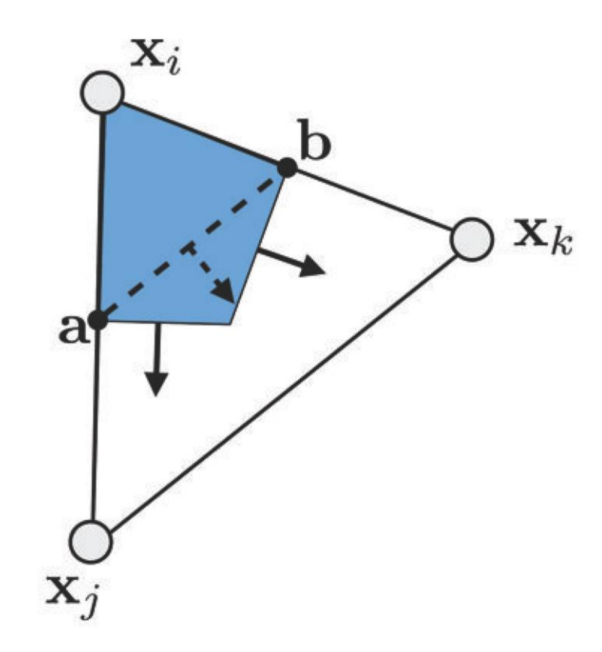
Cotangent Formula

- mixed finite element/finite volume method
  - Assume it constant on each vertex

$$\int_{A_i} \Delta f \, dA = \int_{A_i} \operatorname{div} \nabla f \, dA = \int_{\partial A_i} (\nabla f) \cdot \boldsymbol{n} \, ds$$

1. A_i is the local averaging domain of vertex i.
2. ∂A_i is the boundary of A_i.
3. $\boldsymbol{n}$ is the outward pointing unit normal of the boundary.
4. f is the signal defined on mesh.





Cotangent Formula

We split this integral by considering the integration separately for each triangle T.

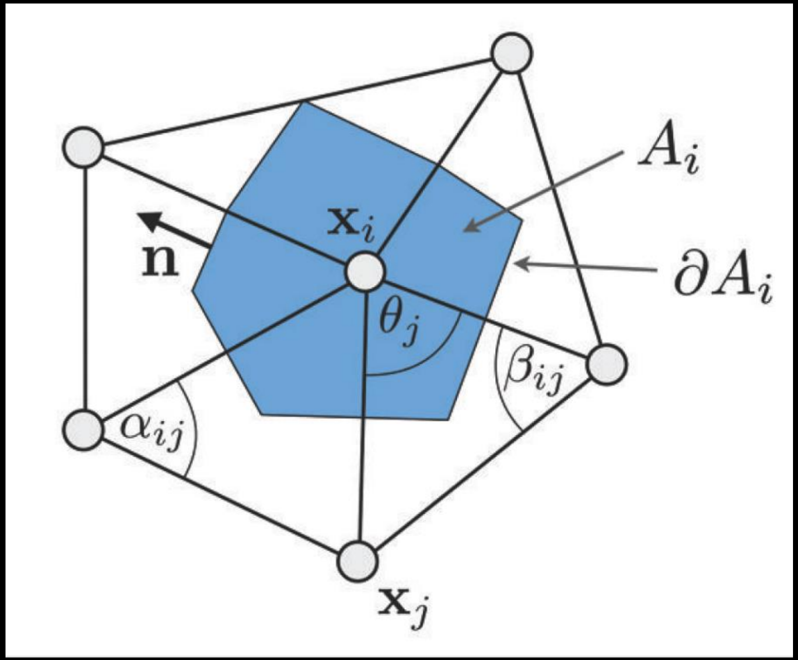
$$\int_{\partial A_i \cap T} \nabla f \cdot \boldsymbol{n} d\boldsymbol{s} = \nabla f \cdot (\boldsymbol{a} - \boldsymbol{b})^{\perp} = \frac{1}{2} \nabla f \cdot \left(\boldsymbol{x}_j - \boldsymbol{x}_k\right)^{\perp}$$

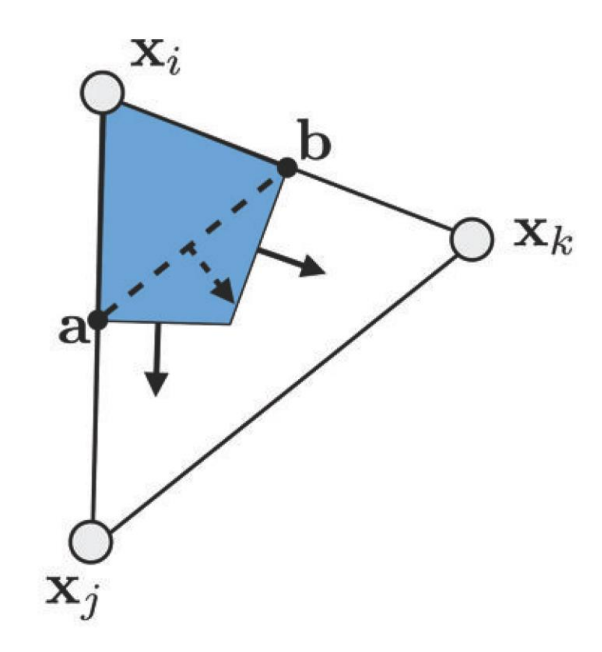
∇f is constant within each triangle.

$$\nabla f = \left(f_j - f_i\right) \frac{(\boldsymbol{x}_i - \boldsymbol{x}_k)^{\perp}}{2A_T} + (f_k - f_i) \frac{\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)^{\perp}}{2A_T}$$

$$\int_{\partial A_i \cap T} \nabla f \cdot \boldsymbol{n} d\boldsymbol{s} = \left(f_j - f_i\right) \frac{(\boldsymbol{x}_i - \boldsymbol{x}_k)^{\perp} \cdot \left(\boldsymbol{x}_j - \boldsymbol{x}_k\right)^{\perp}}{4A_T}$$

$$+ (f_k - f_i) \frac{\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)^{\perp} \cdot \left(\boldsymbol{x}_j - \boldsymbol{x}_k\right)^{\perp}}{4A_T}$$





Cotangent Formula

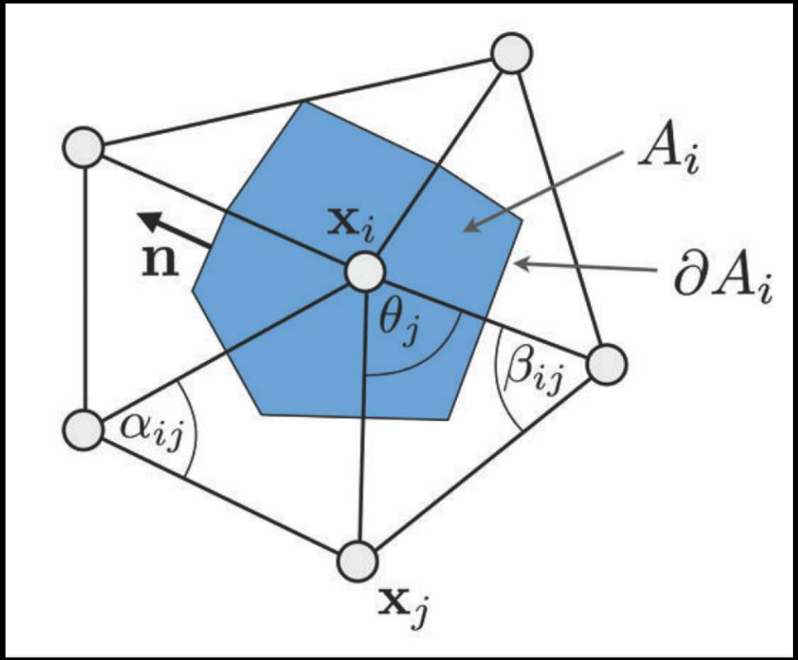
- Because:

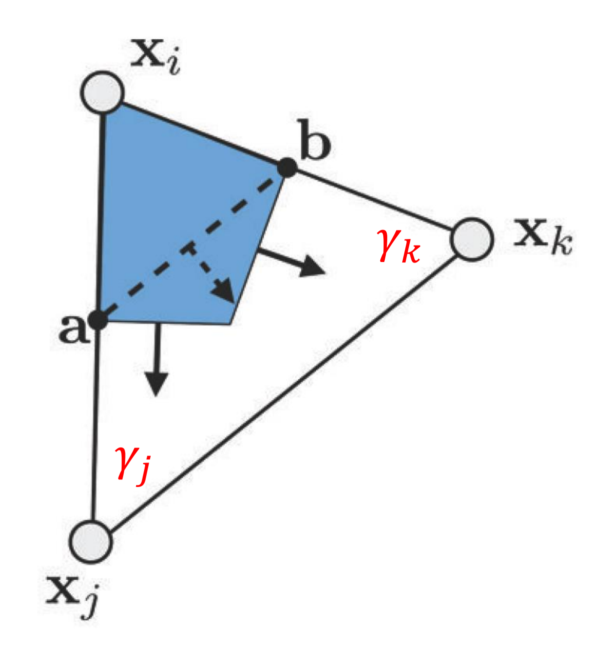
$$A_T = \frac{1}{2}\sin\gamma_j \|\boldsymbol{x}_j - \boldsymbol{x}_i\| \, \|\boldsymbol{x}_j - \boldsymbol{x}_k\|$$
$$= \frac{1}{2}\sin\gamma_k \|\boldsymbol{x}_i - \boldsymbol{x}_k\| \, \|\boldsymbol{x}_j - \boldsymbol{x}_k\|$$

and

$$\cos\gamma_j = \frac{(\boldsymbol{x}_j - \boldsymbol{x}_i)\cdot(\boldsymbol{x}_j - \boldsymbol{x}_k)}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|\|\boldsymbol{x}_j - \boldsymbol{x}_k\|}$$

$$\cos\gamma_k = \frac{(\boldsymbol{x}_i - \boldsymbol{x}_k)\cdot(\boldsymbol{x}_j - \boldsymbol{x}_k)}{\|\boldsymbol{x}_i - \boldsymbol{x}_k\|\|\boldsymbol{x}_j - \boldsymbol{x}_k\|}$$



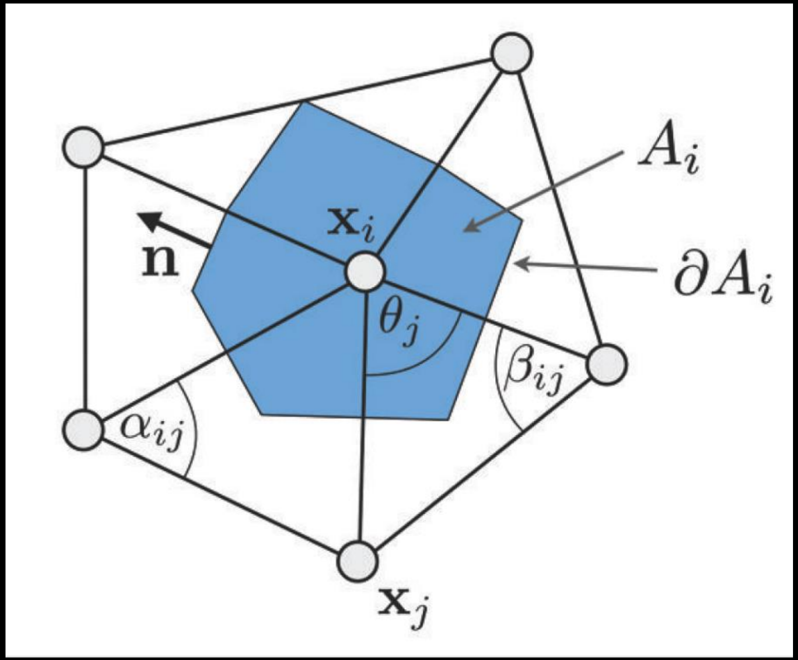


Cotangent Formula

- and

$$(\boldsymbol{x}_i - \boldsymbol{x}_k)^{\perp} \cdot (\boldsymbol{x}_j - \boldsymbol{x}_k)^{\perp} = (\boldsymbol{x}_i - \boldsymbol{x}_k) \cdot (\boldsymbol{x}_j - \boldsymbol{x}_k)$$

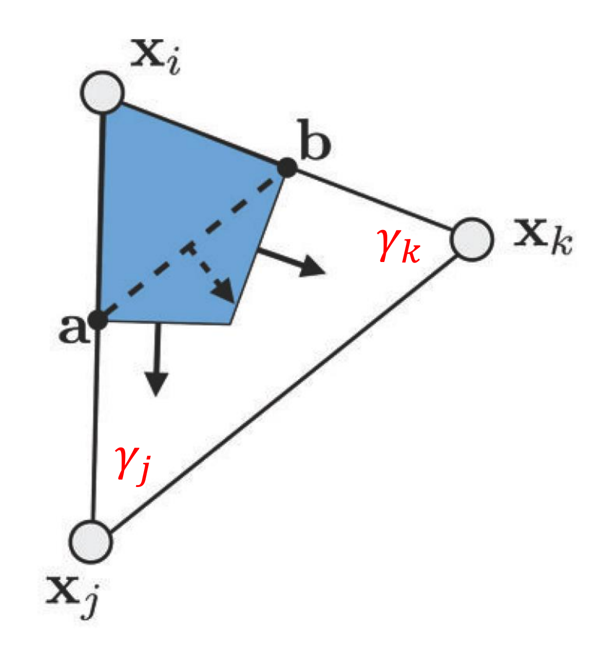
$$(\boldsymbol{x}_j - \boldsymbol{x}_i)^{\perp} \cdot (\boldsymbol{x}_j - \boldsymbol{x}_k)^{\perp} = (\boldsymbol{x}_j - \boldsymbol{x}_i) \cdot (\boldsymbol{x}_j - \boldsymbol{x}_k)$$



So

$$\int_{\partial A_i \cap T} \nabla f \cdot \boldsymbol{n} ds$$

$$= \frac{1}{2}\left(\cot\gamma_k (f_j - f_i) + \cot\gamma_j (f_k - f_i)\right)$$

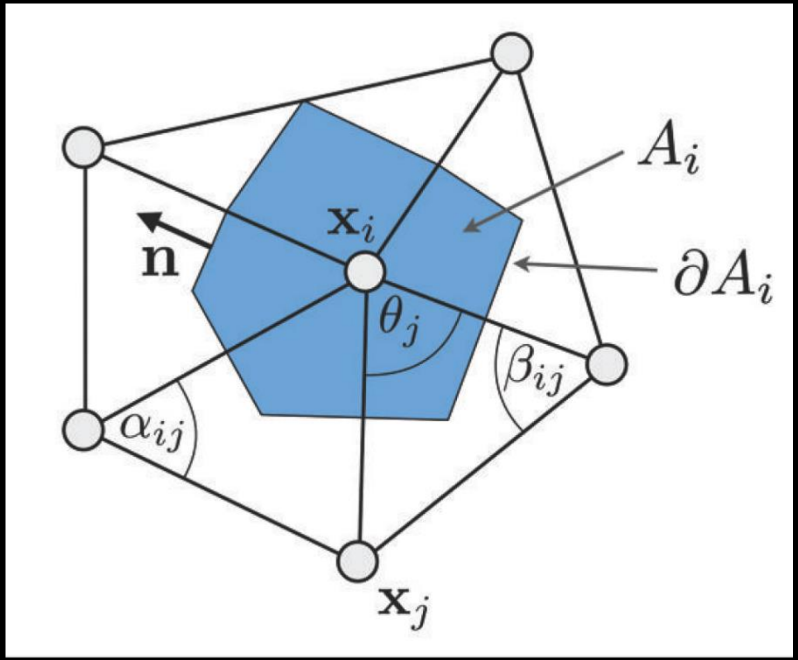


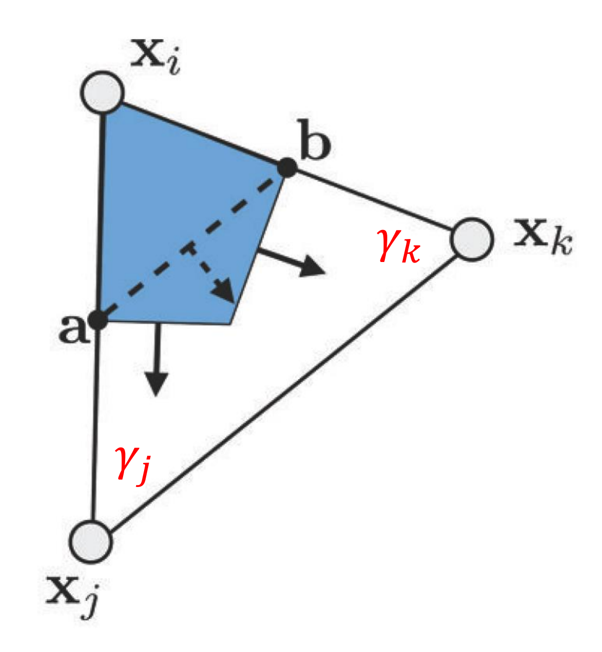
Cotangent Formula

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

Discrete average of the Laplace-Beltrami operator of a function f at vertex v_i is given as:

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





1. most widely used discretization

2. $(\cot\alpha_{ij} + \cot\beta_{ij})$ become negative if $\alpha_{ij} + \beta_{ij} > \pi$. Violate the property of POSITIVE WEIGHTS.

No free lunch

- Main result Not all meshes admit Laplacians satisfying properties (SYMMETRY), (LOCALITY), (LINEAR PRECISION), and (POSITIVE WEIGHTS) simultaneously.

Implementation thinking

- How to compute the cotangent formula?
- One simple idea: for each edge, compute the related cot value.

- Any improvement? More efficient?

Discrete Curvature

- When applied to the coordinate function $\boldsymbol{x}$, the Laplace-Beltrami operator provides a discrete approximation of the mean curvature normal.

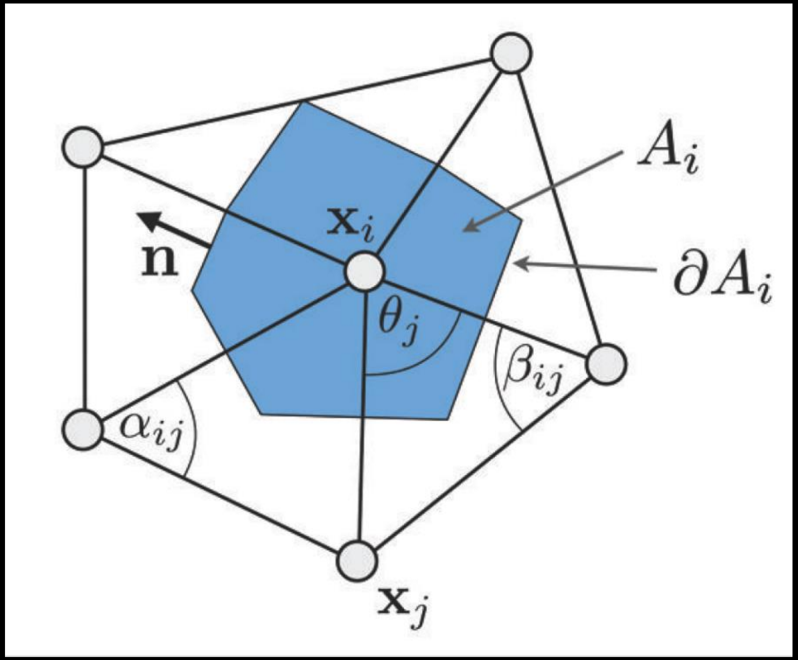
$$\Delta \boldsymbol{x} = -2H\boldsymbol{n}$$

absolute discrete mean curvature at vertex i:

$$H_i = \frac{1}{2} ||\Delta \boldsymbol{x}||$$

- A discrete operator for Gaussian curvature:

$$K_i = \frac{1}{A_i}\left(2\pi - \sum_{j\in\Omega(i)} \theta_j\right)$$



Implementation thinking

- How to compute the discrete Gaussian curvature?
- One simple idea: for each vertex, compute the related angle.

- Any improvement? More efficient?

Second homework

- Color bar
  - Map a value to a color
- Visualize:
  - mean curvature,
  - absolute mean curvature,
  - and Gaussian curvature.