1 Problem Setup

As input, we are given a graph $G = (V_G, E_G)$, where each vertex is a geographic position $s_i \in S^2$, and each edge (i, j) has an associated (Olivier-Ricci) curvature $R_{i,j} \in (-2, 1)$ and an associated latency $t_{i,j} \in \mathbb{R}_{\geq 0}$.

Intuitively, we want to return a surface in $\mathbb{R}^3$ that is the graph of a function $\rho : S^2 \to \mathbb{R}_{>0}$ whose geodesics $g_{i,j}$ between s_i and s_j (and their missing ρ-coordinates) have length $\phi_{i,j}$ that is in a linear relationship with the latency.

The strategy to realize this intuition is to create a mesh $M = (V_M, E_M)$ supported on a subset of S^2 that contains our input positions. Let P be the support. Then for each $s_i \in P$, we want to assign a $\rho_i \in \mathbb{R}_{>0}$, which in turn gives a point $v_i = (s_i, \rho_i) \in V$. This setup is made explicit in `mesh/sphere.py`.

A similar setup is found in `mesh/rectangle.py`, where we use $[0, 1]^2$ instead of S^2. In general, this setup

2 Objective/Loss Functions

To enforce that the mesh approximates our desired surface, we define the objective functions

$$\begin{aligned}
\mathcal{L}_{\text{geodesic}}(M) &\triangleq \sum_{e \in E_G} (\text{least squares residual of edge } e)^2, \\
\mathcal{L}_{\text{smooth}}(M) &\triangleq -\rho^\intercal L_C \rho, \\
\mathcal{L}_{\text{curvature}}(M) &\triangleq \sum_{\substack{v \in V_M \\ v \text{ close to } (i,j)}} \left(\kappa(v) - R_{i,j}\right)^2, \\
\mathcal{L}(M) &\triangleq \lambda_{\text{geodesic}} \mathcal{L}_{\text{geodesic}}(M) + \lambda_{\text{curvature}} \mathcal{L}_{\text{curvature}}(M) + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}}(M),
\end{aligned}$$

where the λ's are tunable hyperparameters. The other variables will be defined in the upcoming subsections. Our goal is then to minimize $\mathcal{L}(M)$.

Note that the loss functions (particularly the geodesic and total ones) also have a dependence on the measured latencies. We omit that as a written parameter because they are treated as fixed (we are really optimizing over the manifold, not over the measured latencies).

2.1 Laplacian

Some mesh notation first. If i and j are two indices vertices for which (v_i, v_j) is a half-edge, let $c(i, j)$ be the index such that $v_i \to v_j \to v_{c(i,j)}$ traces a triangle counterclockwise. Note that this index exists and is unique assuming we have a mesh without boundary. On a mesh with boundary, if no $c(i, j)$ exists, then the half-edge (v_i, v_j) lies on the boundary.

We define the following variables:

$N_{i,j}$	Outward normal of triangle $v_i \to v_j \to v_{c(i,j)}$
$A_{i,j}$	Area of triangle $v_i \to v_j \to v_{c(i,j)}$
$D_{i,j}$	Vertex triangle areas; diagonal
$\theta_{i,j}$	Measure of $\angle v_i v_{c(i,j)} v_j$
L_C^{Neumann}	Cotangent operator with zero-Neumann boundary condition
$L_C^{\text{Dirichlet}}$	Cotangent operator with zero-Dirichlet boundary condition
L_C	Cotangent operator in the no-boundary case; sparse

2.1.1 Forward Computation

We have the following (standard) definition of the Laplace-Beltrami operator on a mesh:

$$\begin{aligned}
N_{i,j} &= \left(v_i - v_{c(i,j)}\right) \times \left(v_j - v_{c(i,j)}\right), \\
A_{i,j} &= \frac{1}{2} \left\| N_{i,j} \right\|_2, \\
D_{i,j} &= \begin{cases} \dfrac{1}{3} \displaystyle\sum_{\substack{k \\ (v_i, v_k) \text{ is} \\ \text{a half-edge}}} A_{i,k} & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases}
\end{aligned}$$

$$
\begin{aligned}
\cot\bigl(\theta_{i,j}\bigr) &= \frac{\Bigl(v_i - v_{c(i,j)}\Bigr) \cdot \Bigl(v_j - v_{c(i,j)}\Bigr)}{2A_{i,j}}, \\
\Bigl(L_C^{\text{Neumann}}\Bigr)_{i,j} &= \begin{cases} \frac{1}{2}\Bigl(\cot\bigl(\theta_{i,j}\bigr) + \cot\bigl(\theta_{j,i}\bigr)\Bigr) & \text{if } \bigl(v_i, v_j\bigr) \text{ or } \bigl(v_j, v_i\bigr) \text{ is a half-edge}, \\ -\frac{1}{2} \displaystyle\sum_{\substack{k \\ (v_i, v_k) \text{ or } (v_k, v_i) \\ \text{is a half-edge}}} \Bigl(\cot\bigl(\theta_{i,k}\bigr) + \cot\bigl(\theta_{k,i}\bigr)\Bigr) & \text{if } i = j, \\ 0 & \text{otherwise}, \end{cases} \\
\Bigl(L_C^{\text{Dirichlet}}\Bigr)_{i,j} &= \begin{cases} \frac{1}{2}\Bigl(\cot\bigl(\theta_{i,j}\bigr) + \cot\bigl(\theta_{j,i}\bigr)\Bigr) & \text{if } \bigl(v_i, v_j\bigr) \text{ and } \bigl(v_j, v_i\bigr) \text{ are half-edges}, \\ -\frac{1}{2} \displaystyle\sum_{\substack{k \\ (v_i, v_k) \text{ and } (v_k, v_i) \\ \text{are half-edges}}} \Bigl(\cot\bigl(\theta_{i,k}\bigr) + \cot\bigl(\theta_{k,i}\bigr)\Bigr) & \text{if } i = j, \\ 0 & \text{otherwise}. \end{cases}
\end{aligned}
$$

Flipping our attention back to meshes without boundary, the two definitions above coincide, so we can write

$$L_C = L_C^{\text{Neumann}} = L_C^{\text{Dirichlet}}.$$

2.1.2 Reverse Computation

For the ease of notation, assume that we are using the spherical setup, so $v_\ell = \rho_\ell s_\ell$.

We compute

$$
\begin{aligned}
\frac{\partial v_i}{\partial \rho_\ell} &= \begin{cases} s_i & \text{if } \ell = i, \\ 0 & \text{otherwise}, \end{cases} \\
\frac{\partial N_{i,j}}{\partial \rho_\ell} &= \begin{cases} \Bigl(v_{c(i,j)} - v_j\Bigr) \times \frac{\partial v_\ell}{\partial \rho_\ell} & \text{if } \ell = i, \\ \Bigl(v_i - v_{c(i,j)}\Bigr) \times \frac{\partial v_\ell}{\partial \rho_\ell} & \text{if } \ell = j, \\ \bigl(v_j - v_i\bigr) \times \frac{\partial v_\ell}{\partial \rho_\ell} & \text{if } \ell = c(i,j), \\ 0 & \text{otherwise}, \end{cases} \\
\frac{\partial A_{i,j}}{\partial \rho_\ell} &= \frac{1}{4A_{i,j}} N_{i,j} \cdot \frac{\partial N_{i,j}}{\partial \rho_\ell}, \\
\left(\frac{\partial D}{\partial \rho_\ell}\right)_{i,j} &= \begin{cases} \frac{1}{3} \displaystyle\sum_{\substack{k \\ (v_i, v_k) \text{ is} \\ \text{a half-edge}}} \frac{\partial A_{i,k}}{\partial \rho_\ell} & \text{if } i = j, \\ 0 & \text{otherwise}, \end{cases} \\
\frac{\partial}{\partial \rho_\ell} \cot\bigl(\theta_{i,j}\bigr) &= \begin{cases} \dfrac{\Bigl(v_j - v_{c(i,j)}\Bigr) \cdot \frac{\partial v_\ell}{\partial \rho_\ell} - 2\cot\bigl(\theta_{i,j}\bigr)\frac{\partial A_{i,j}}{\partial \rho_\ell}}{2A_{i,j}} & \text{if } \ell = i, \\ \dfrac{\Bigl(v_i - v_{c(i,j)}\Bigr) \cdot \frac{\partial v_\ell}{\partial \rho_\ell} - 2\cot\bigl(\theta_{i,j}\bigr)\frac{\partial A_{i,j}}{\partial \rho_\ell}}{2A_{i,j}} & \text{if } \ell = j, \\ \dfrac{\Bigl(2v_{c(i,j)} - v_i - v_j\Bigr) \cdot \frac{\partial v_\ell}{\partial \rho_\ell} - 2\cot\bigl(\theta_{i,j}\bigr)\frac{\partial A_{i,j}}{\partial \rho_\ell}}{2A_{i,j}} & \text{if } \ell = c(i,j), \\ 0 & \text{otherwise}, \end{cases}
\end{aligned}
$$

$$\left(\frac{\partial L_C^{\text{Neumann}}}{\partial \rho_\ell}\right)_{i,j} = \begin{cases} \frac{1}{2}\left(\frac{\partial}{\partial \rho_\ell}\cot(\theta_{i,j}) + \frac{\partial}{\partial \rho_\ell}\cot(\theta_{j,i})\right) & \text{if } (v_i, v_j) \text{ or } (v_j, v_i) \text{ is a half-edge,} \\ -\frac{1}{2} \sum_{\substack{k \\ (v_i, v_k) \text{ or } (v_k, v_i) \\ \text{is a half-edge}}} \left(\frac{\partial}{\partial \rho_\ell}\cot(\theta_{i,k}) + \frac{\partial}{\partial \rho_\ell}\cot(\theta_{k,i})\right) & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases}$$

$$\left(\frac{\partial L_C^{\text{Neumann}}}{\partial \rho_\ell}\right)_{i,j} = \begin{cases} \frac{1}{2}\left(\frac{\partial}{\partial \rho_\ell}\cot(\theta_{i,j}) + \frac{\partial}{\partial \rho_\ell}\cot(\theta_{j,i})\right) & \text{if } (v_i, v_j) \text{ and } (v_j, v_i) \text{ are half-edges,} \\ -\frac{1}{2} \sum_{\substack{k \\ (v_i, v_k) \text{ and } (v_k, v_i) \\ \text{are half-edges}}} \left(\frac{\partial}{\partial \rho_\ell}\cot(\theta_{i,k}) + \frac{\partial}{\partial \rho_\ell}\cot(\theta_{k,i})\right) & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

2.2 Geodesic Distance via the Heat Method

Here are the variables used for this part of the computation:

γ	Set of points in V_M
t	Square of mean half-edge length
δ^γ	Heat source (indicator on γ)
u^γ	Heat flow
$q_{i,j}^\gamma$	Intermediate value for computation
$m_{i,j}^\gamma$	Intermediate value for computation
$X_{i,j}^\gamma$	Unit vector in same direction as $\nabla u_{i,j}^\gamma$
$p_{i,j}^\gamma$	Intermediate value for computation
ϕ^γ	Vector of geodesic distances

2.2.1 Forward Computation

Say we want to find the geodesic distances to a set of points $\gamma \subseteq V_M$. Following the Crane et al's Heat Method, we use the (approximate) heat flow u^γ, where

$$\begin{aligned} t &\triangleq (\text{mean spacing between mesh points})^2, && \text{Adjustable parameter} \\ \delta^\gamma &\triangleq \begin{cases} 1 & \text{if } v_i \in \gamma, \\ 0 & \text{if } v_i \notin \gamma, \end{cases} && \text{Heat source} \\ u^\gamma &\triangleq (D - tL_C)^{-1}\delta^\gamma && \text{Heat flow} \end{aligned}$$

Similar to when computing the Laplacian, we need to be careful about computing these values on a mesh with boundary. Following Crane et al's advice, we compute u^γ using both the zero Neumann and zero Dirichlet boundary conditions, and then average them.

With u^γ in hand, we can then compute

$$\begin{aligned} q_{i,j} &\triangleq u_i^\gamma (v_{c(i,j)} - v_j), \\ m_{i,j} &\triangleq q_{i,j} + q_{j,c(i,j)} + q_{c(i,j),i}, \\ (\widetilde{\nabla} u^\gamma)_{i,j} &\triangleq N_{i,j} \times m_{i,j}, \\ X_{i,j}^\gamma &\triangleq -\frac{(\widetilde{\nabla} u^\gamma)_{i,j}}{\|(\widetilde{\nabla} u^\gamma)_{i,j}\|_2}, \\ p_{i,j} &\triangleq \cot(\theta_{i,j})(v_j - v_i), \\ (\nabla \cdot X^\gamma)_i &= \frac{1}{2} \sum_{\substack{k \\ (v_i, v_k) \text{ is an edge}}} (p_{i,k} - p_{c(i,k),i}) \cdot X_{i,k}^\gamma, \\ \phi^\gamma &= L_C^+ \cdot (\nabla \cdot X^\gamma). \end{aligned}$$

Here, L_C^+ is the pseudoinverse of L_C (as it is singular). Note that the integrated divergence can be thought of as taking a sum over triangles $v_i \to v_k \to v_{c(i,k)}$.

Note that we're being careful about which pieces have a dependence on γ, as we can reuse certain computations if we want to compute distances from multiple sources. Abusing notation, we can get the distance matrix (that is, get rid of the γ dependence) from

$$\phi_{i,j} = \left(\phi^{\{v_j\}}\right)_i .$$

2.2.2 Reverse Computation

Note that $c(i, c(j,i)) = j$. This is helpful for reindexing some sums (in particular, the one for $\nabla \cdot X$).

We then have the following partial derivatives:

$$\begin{aligned}
\frac{\partial u^\gamma}{\partial \rho_\ell} &= -(D - tL_C)^{-1}\left(\frac{\partial D}{\partial \rho_\ell} - t\frac{\partial L_C}{\partial \rho_\ell}\right)u^\gamma, \\
\frac{\partial q_{i,j}}{\partial \rho_\ell} &= \begin{cases} \frac{\partial u_i^\gamma}{\rho_\ell}(v_{c(i,j)} - v_j) - u_i^\gamma \frac{\partial v_\ell}{\rho_\ell} & \text{if } \ell = j, \\ \frac{\partial u_i^\gamma}{\rho_\ell}(v_{c(i,j)} - v_j) + u_i^\gamma \frac{\partial v_\ell}{\partial \rho_\ell} & \text{if } \ell = c(i,j), \\ \frac{\partial u_i^\gamma}{\rho_\ell}(v_{c(i,j)} - v_j) & \text{otherwise,} \end{cases} \\
\frac{\partial m_{i,j}}{\partial \rho_\ell} &= \frac{\partial q_{i,j}}{\partial \rho_\ell} + \frac{\partial q_{j,c(i,j)}}{\partial \rho_\ell} + \frac{\partial q_{c(i,j),i}}{\partial \rho_\ell}, \\
\frac{\partial (\widetilde{\nabla} u^\gamma)_{i,j}}{\partial \rho_\ell} &= \frac{\partial N_{i,j}}{\partial \rho_\ell} \times m_{i,j} + N_{i,j} \times \frac{\partial m_{i,j}}{\partial \rho_\ell}, \\
\frac{\partial X_{i,j}^\gamma}{\partial \rho_\ell} &= -\frac{1}{\|(\widetilde{\nabla} u^\gamma)_{i,j}\|_2}(I - X_{i,j}^\gamma (X_{i,j}^\gamma)^\intercal)\frac{\partial (\widetilde{\nabla} u^\gamma)_{i,j}}{\partial \rho_\ell}, \\
\frac{\partial p_{i,j}}{\partial \rho} &= \begin{cases} \left(\frac{\partial}{\partial \rho_\ell}\cot(\theta_{i,j})\right)(v_j - v_i) - \cot(\theta_{i,j})\frac{\partial v_\ell}{\rho_\ell} & \text{if } \ell = i, \\ \left(\frac{\partial}{\partial \rho_\ell}\cot(\theta_{i,j})\right)(v_j - v_i) + \cot(\theta_{i,j})\frac{\partial v_\ell}{\rho_\ell} & \text{if } \ell = j, \\ \left(\frac{\partial}{\partial \rho_\ell}\cot(\theta_{i,j})\right)(v_j - v_i) & \text{if } \ell = c(i,j), \\ 0 & \text{otherwise,} \end{cases} \\
\frac{\partial (\nabla \cdot X^\gamma)_i}{\partial \rho_\ell} &= \frac{1}{2}\sum_{\substack{k \\ (v_i, v_k) \text{ is an edge}}} \left(\left(\frac{\partial p_{i,k}}{\partial \rho_\ell} - \frac{\partial p_{c(i,k),i}}{\partial \rho_\ell}\right)\cdot X_{i,k}^\gamma + (p_{i,k} - p_{c(i,k),i})\cdot \frac{\partial X_{i,k}^\gamma}{\partial \rho_\ell}\right) \\
\frac{\partial \phi^\gamma}{\partial \rho_\ell} &= L_C^+\left(\frac{\partial (\nabla \cdot X^\gamma)}{\partial \rho_\ell} - \frac{\partial L_C}{\partial \rho_\ell}\phi^\gamma\right).
\end{aligned}$$