Heat Equation With Backward Euler (Solving Linear Systems)

We will solve heat equation

$$u' = \alpha \Delta u \tag{1}$$

using backward Euler.

We use the discrete Laplacian operator for triangle meshes:

$$\Delta u_i = \frac{1}{2} \sum_j (\cot \alpha_{ij} + \cot \beta_{ij})(u_j - u_i) \tag{2}$$

Applying backward Euler to Eq(1), we get for each vertex i:

$$u_i^{k+1} = u_i^k + \tau\alpha\Delta u_i^{k+1} \tag{3}$$

Substituting Eq(2) into Eq(3), we have

$$\frac{\tau\alpha}{2} \sum_j (\cot \alpha_{ij} + \cot \beta_{ij}) u_j^{k+1} - \left[\frac{\tau\alpha}{2} \sum_j (\cot \alpha_{ij} + \cot \beta_{ij}) + 1 \right] u_i^{k+1} = -u_i^k \tag{4}$$

We assume that from timestep k to timestep $k+1$, α_{ij} and β_{ij} do not change much. Thus, we use α_{ij} and β_{ij} at timestep k in Eq(4).

Eq(4) is a linear system, which can be assembled into matrix form and passed into a linear system solver. We assemble all unknowns in the following form:

$$x = \begin{bmatrix} u_0^{k+1} \\ \vdots \\ u_{n-1}^{k+1} \end{bmatrix} \tag{5}$$

where n is the number of vertices in the mesh. The coefficients can be assembled into an n-by-n matrix A:

$$A(i,p) = \begin{cases} \frac{\tau\alpha}{2}(\cot \alpha_{ip} + \cot \beta_{ip}) & \text{if } p \text{ is a neighbor of } i \\ -\frac{\tau\alpha}{2} \sum_j (\cot \alpha_{ij} + \cot \beta_{ij}) - 1 & \text{if } p = i \\ 0 \quad \text{otherwise} \end{cases} \tag{6}$$

Since a vertex will usually have a few neighbors in a mesh, A will be a sparse matrix.

We also need b on the right side of the linear system. b is constructed by extracting the right hand side of Eq(4) and put them into corresponding positions:

$$b = \begin{bmatrix} -u_0^k \\ \vdots \\ -u_{n-1}^k \end{bmatrix} \tag{7}$$

After everything is assembled, we only need to solve $Ax = b$ for x using Eigen package. We have tried different linear solvers in Eigen package and decided to use `LeastSquaresConjugateGradient` solver on sparse matrix for the purpose of numerical stability. Specifically, assume that we have a sparse matrix A. There is no guarantee that A is a positive semi-definite (PSD) matrix. However, $A^T A$ is guaranteed to be PSD. Then we need to solve

$$A^T A x = A^T b \tag{8}$$

using `LeastSquaresConjugateGradient`. The result is

$$x = (A^T A)^{-1} A^T b \tag{9}$$

where A, b are known values.