Material

Definition of uniform and cotangent Laplacian equations is found in the parametrization presentation slides 24 to 28.

The Constraints Matrix

In the assignment, it is written that the linear system of equations of the boundary constraints, is defined as:

$$C\begin{pmatrix} u \\ v \end{pmatrix} = d$$

Please pay attention that the column vector $\begin{pmatrix} u \\ v \end{pmatrix}$ is a concatenation of the two column vectors u and v, where:

$$u = [u_1 \quad u_2 \quad \dots \quad u_{n-1} \quad u_n]^T$$
$$v = [v_1 \quad v_2 \quad \dots \quad v_{n-1} \quad v_n]^T$$

Given that our mesh has n vertices to be parameterized (that is, if our mesh is defined as (V, F), then $n = |V|$).

Therefore, the number of columns of C is $2n$. However, the number of rows of C is dynamic, and depends on the number of boundary vertices of our mesh (which we can get by **igl:: boundary_loop**).

For example, let's say that our mesh has 6 vertices in total, and that its boundary is consisted of 3 vertices. Let's say that the boundary vertices are p_3, p_4, p_5 (we have found them using **igl:: boundary_loop**). Now we want to map these boundary vertices into the plane, such that they will be located on the perimeter of the unit circle. In order to do it, we use **igl:: map_boundary_to_circle** (in order to understand how it is done, you can read the documentation of this function).

Let's say that we have found out that:

$$(u_3, v_3) = (d_1, d_4)$$
$$(u_4, v_4) = (d_2, d_5)$$
$$(u_5, v_5) = (d_3, d_6)$$

Therefore, we can now build the matrix C, which will look as follows:

$$C\begin{pmatrix}u\\v\end{pmatrix}=d\Rightarrow\begin{pmatrix}0&0&1&0&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&0&0&1&0\end{pmatrix}\begin{pmatrix}u_1\\u_2\\u_3\\u_4\\u_5\\u_6\\v_1\\v_2\\v_3\\v_4\\v_5\\v_6\end{pmatrix}=\begin{pmatrix}d_1\\d_2\\d_3\\d_4\\d_5\\d_6\end{pmatrix}$$

As you can see, we have got that C is a sparse indicator matrix, which is not necessarily square.

The System of Equations to Be Solved

Once have found the constraints matrix, wee use thee following system of equations to find the coordinates of the internal non-boundary vertices:

$$\begin{pmatrix}A&C^T\\C&0\end{pmatrix}\begin{pmatrix}u\\v\\\lambda\end{pmatrix}=\begin{pmatrix}b\\d\end{pmatrix}$$

Where:

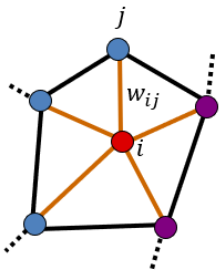
- Again, u and v are both column vectors as shown above – these vectors are our **actual unknowns** which we want to find solution for. We do know the values of entries of u and v which corresponds to the constrained boundary points.
- C is the constraints matrix (we know it, as shown above).
- d is the vector of positions of the constrained points (we know it, as shown above).
- All we're left with, is to define A and b, which both depends on the parametrization method that we will use. The equation $A\begin{pmatrix}u\\v\end{pmatrix}=b$ will describe the requirements that we impose on the non-boundary points in order to achieve the desired parameterization.
- Pay attention, the size of A will always be $2n\times 2n$ (that is, $2|V|\times 2|V|$). It can be easily be deduced by the final matrix form of the system of linear equations above.
- λ is also a vector of the same size as d. Once you will find a solution for the linear system of equation above, wee will find thee solution vector $\begin{pmatrix}u\\v\\\lambda\end{pmatrix}$ – we will be only interested in the $\begin{pmatrix}u\\v\end{pmatrix}$ part of that vector. You can ignore λ.
- For more information of how this form of system of linear equations was derived (if you wish to know), please see this presentation.

Uniform and Cotangent Laplacian Solution Approach

Once we have found the fixed positions of the boundary vertices, we want to find the coordinates of the internal non-boundary vertices. We would like that each non-boundary parametrization vertex $p_i = (u_i, v_i)$ will satisfy that $\Delta(p_i) = 0$. That is, we would like that its Laplacian will be equal to 0.

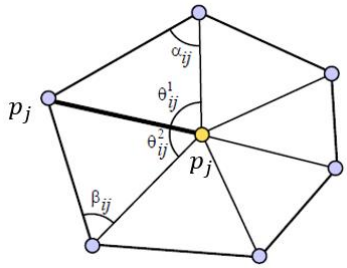
Since we're dealing with polygonal meshes, and not with continuous manifolds, we use the discrete version of the Laplacian operator:

$$\Delta(p_i) = \frac{1}{W_i} \sum_{j \in N(i)} w_{ij}(p_i - p_j)$$



In the **Uniform Laplacian** case, we use $w_{ij} = 1$ and $W_i = |N(i)|$.

In the **Cotangent Laplacian** case, we use $w_{ij} = \frac{cot(\alpha_{ij}) + cot(b_{ij})}{2}$ and $W_i = voronoi\ cell\ area$.



So, in each of those two cases, we would like that any non-boundary p_i will satisfy:

$$\Delta(p_i) = 0 \Rightarrow \frac{1}{W_i} \sum_{j \in N(i)} w_{ij}(p_i - p_j) = 0$$

Which can be broken into two separate equations:

$$\frac{1}{W_i} \sum_{j \in N(i)} w_{ij}(u_i - u_j) = 0$$

$$\frac{1}{W_i} \sum_{j \in N(i)} w_{ij}(v_i - v_j) = 0$$

Use this set of equations to build the matrix A.

Uniform Laplacian

In the uniform Laplacian case, we get that:

$$\Delta(p_i) = \frac{1}{|N(i)|}\sum_{j\in N(i)} p_i - p_j = p_i - \frac{1}{|N(i)|}\sum_{j\in N(i)} p_j$$

Therefore, we require that:

$$u_i - \frac{1}{|N(i)|}\sum_{j\in N(i)} u_j = 0$$

$$v_i - \frac{1}{|N(i)|}\sum_{j\in N(i)} v_j = 0$$

Both equations can be written in matrix form, as follows:

$$(I - D^{-1}B)u = 0$$

$$(I - D^{-1}B)v = 0$$

Where:

- $N(i)$ is the valence (degree) of vertex i.
- D is a diagonal matrix $D_{ii} = |N(i)|$
- B is the adjacency matrix $B_{ij} = \begin{cases} 1 & (p_i, p_j) \in E \\ 0 & else \end{cases}$
- $u = [u_1 \quad u_2 \quad \dots \quad u_{n-1} \quad u_n]^T$
- $v = [v_1 \quad v_2 \quad \dots \quad v_{n-1} \quad v_n]^T$

That's why we need to use **igl::adjacency_matrix**.

(for more information, see the first two pages of this paper).

Try to think how to build the matrix A by blocks of $I - D^{-1}B$ (you can use **igl::cat** to concatenate matrix blocks into a new matrix).

Cotangent Laplacian

It is even easier – just call **igl::cotmatrix**, and use its returned matrix (which is the cotangent Laplacian matrix itself) to build the matrix A by blocks of the cotangent Laplacian matrix (similar to the last step of the uniform case).