Exercise 5: Laplacians

Throughout this exercise, the term “Laplacian” refers to Laplacian applied to coordinate function.

Combinatorial Laplacian (2 points)

Computing the combinatorial Laplacian is not difficult, particularly when the VV function is available which provides the neighbors of each vertex. Try implementing the Tutte discretization using the following formula:

$$\Delta p(v_i) = \frac{1}{n}\sum_{j=1}^{n}(v_j - v_i) = (\frac{1}{n}\sum_{j=1}^{n} v_j) - v_i$$

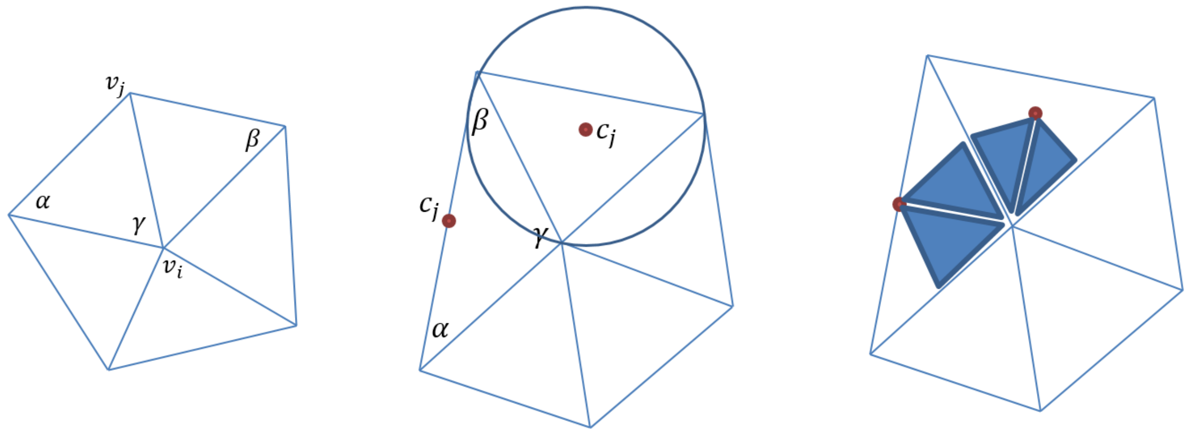
Laplacian provides a vector for each vertex, try visualizing them. Having the array of vectors computed, assign them to the `Vectors` field of the `TriangleMesh` instance and notify the renderer that vectors should be displayed by setting the `ShowVectorAttribute` of the renderer instance to `true`. You can then display the vectors by pressing N in the renderer. Have a look at the result and consider the similarities and differences with respect to an expected mean curvature normal vector.

Cotan Laplacian (3 points)

The “cotan formula”, used for evaluation of the Laplace operator, is of key importance in a number of mesh processing applications. In this exercise, we are going to compute the Laplacian of the coordinate function **p**, which for each vertex returns its coordinates. We are going to evaluate it for each vertex, one by one, i.e. we are not going to construct a matrix representation of the Laplacian. The Laplacian of **p** will be interpreted as a 3D vector assigned to each vertex, and we are going to visualize these vectors using the provided framework.

Having a vertex v_i and its neighbors $v_j, j = 1..n$, the cotan Laplacian of the coordinate function at the vertex location is given as

$$\Delta p(v_i) = \frac{1}{A_i}\sum_{j=1}^{n} w_{ij}(v_j - v_i)$$



Where $w_{ij} = \frac{1}{2}(\cot(\alpha) + \cot(\beta))$ and A_i is the area of influence of the vertex v_i. For the implementation, following steps are necessary for each vertex:

1) Find all neighbors of v_i using the VV function from one of the previous exercises
2) Compute the angles α_i and β_i for $i = 1..n$
3) For each neighboring triangle
    a. compute its angle γ. If either $\alpha > \pi/2$, $\beta > \pi/2$ or $\gamma > \pi/2$ (i.e. if the triangle is obtuse), then c_j is chosen as middle point of the edge opposite to the obtuse angle. Otherwise, c_j is chosen as circumcenter of the triangle
    b. compute the contribution of the triangle to A_i. It consists of areas of two triangles, each consisting of v_i, c_j and the centerpoint of one of the triangle edges
4) using the angles and areas, compute the value of the Laplacian of p at the vertex.

Checking correctness

Since $\Delta p = -2Hn$, it easy to check whether the obtained values are correct on an example where the curvature is known. For example, a sphere of radius 1 has both principal curvatures equal to 1 at all vertices, i.e. H = 1 at all vertices. Therefore, the resulting vectors should be normal to the surface and have a length of 2, i.e. they should just touch the surface on the other side of the sphere.

Notes

$$\cot(\alpha) = \frac{1}{\tan(\alpha)}$$

Computation of triangle circumcentre

Having a triangle (v_1, v_2, v_3) with edge lengths $a = \|v_2 - v_3\|$, $b = \|v_3 - v_1\|$ and $c = \|v_2 - v_1\|$, the barycentric coordinates of the circumcentre are

$$d_1 = a^2(b^2 + c^2 - a^2),$$

$$d_2 = b^2(c^2 + a^2 - b^2),$$

$$d_3 = c^2(a^2 + b^2 - c^2).$$

Note that it is necessary to normalize the barycentric coordinates, so that $d_1 + d_2 + d_3 = 1$. Finally, the circumcentre is computed as $c_i = d_1 v_1 + d_2 v_2 + d_3 v_3$. You should check the correctness of the circumcentre computation by evaluating the distance from the circumcentre to the triangle vertices.