Digital Geometry Processing (236329)
Homework #3

Due date: 14/12/2022

Introduction

In this assignment you will compute discrete differential geometry operators and curvature quantities. Moreover, you will perform some analysis on the resulting differential operators. The report should contain demonstrations of all the tasks. Submit your report and your MATLAB code.

Objectives

1. Vector field visualization: implement a function which allows the user to visualize general vector fields on surfaces. Your code should support visualizing a vector field on top of a function (functions are visualized using color, vector fields using arrows). **Tip**: hold on/off, quiver3.

2. Discrete differential operators:
a. Follow the constructions we showed in class and implement functions which take a mesh and return the differential operators grad, div and L. Add to your report snapshots of the gradient and divergence of several functions and vector fields for various meshes.
b. Implement a function that computes the Laplacian directly through sparse construction with cot of angles, and compare with the operator based construction.
c. Compute (analytically) the cot Laplacian of a triangulated quad grid. Then load the mesh "tri_quad_grid" and verify that the computation matches your expectations.

3. Vertex normals: compute a normal field on vertices $n_{\mathcal{V}}$ through the normalization of $\tilde{n}_{\mathcal{V}}(i) = \sum_j A_{\mathcal{F}(j)} n_{\mathcal{F}}(j)$, where i is a vertex and the sum runs over the neighboring faces with associated areas $A_{\mathcal{F}}$ and normals $n_{\mathcal{F}}$. Namely, $n_{\mathcal{V}}(i) = \tilde{n}_{\mathcal{V}}(i)/\|\tilde{n}_{\mathcal{V}}(i)\|$. Show example results in your report.

4. Mean and Gaussian curvatures: implement functions which compute the mean and Gaussian curvatures. Verify your code by using it on a couple of meshes and add the results to the report. It is recommended to test your code on spheres, where the curvature can be computed analytically (please show in your report results for non trivial meshes as well).

Analysis

1. Properties of the cotangent weights matrix: $W = \frac{1}{4} E^T G_{\mathcal{F}}^{-1} E$ is the cotangent weights matrix. Show the following properties of W (you may refer to the paper "Discrete Laplace operators: No free lunch" by Wardetzky et al. for more details, but it is not a must):

- $(NULL)$ $Wf = 0$, whenever f is a constant function, e.g., compute the norm of Wf for various meshes and constant functions, attach the results to your report.
- (SYM) W is a symmetric matrix, e.g., compute $\|W - W^T\|_F$ for various meshes, where $\|\cdot\|_F$ is the Frobenius norm.

- (LOC) W is sparse, e.g., compute the non-zero elements of W. How is this value related to the number of edges $|\mathcal{E}|$?
- (POS) Unfortunately, W is **not** positive. Show it for various meshes. **Tip**: which triangles have negative cotangent weights?
- (PSD) W is positive semi-definite and you should show it for a couple of meshes. **Tip**: eigs with parameters ‘sm’ and ‘lm’.

The paper describes additional properties (LIN) and (CON), but you can ignore them for now.

2. Representation using a reduced basis: in this section we will investigate how discrete functions can be approximated using a reduced smooth basis. To this end, you will compute bases for functions of different sizes k_i and then, you will project a function onto this basis and measure the error as a function of k_i. Specifically, you should implement the following,
   - A smooth basis: given a positive scalar k_i, compute the eigendecomposition of W to k_i eigenvectors. We denote the result by $B_i \in \mathbb{R}^{|\mathcal{V}| \times k_i}$. **Tip**: eigs with ‘sm’ and k_i. In your report, please include a snapshot of the first 9 eigenfunctions corresponding to the lowest eigenvalues on a couple of meshes.
   - Take a “hard-to-represent” function f, e.g., a hat function (1 in one vertex and zero everywhere else) or the eigenvector corresponding to the biggest eigenvalue.
   - Project f onto B_i and back: this can be achieved simply by computing $g_i = B_i(B_i^T f)$, where the parentheses are *important*.
   - Compute the norm of $g_i - f$.
   - Repeat the process for various k_i, e.g., spanned linearly between 10 to 300 with jumps of 20. Add the resulting graph of pairs $(k_i, \|g_i - f\|)$ to your report with snapshots of the desired function and some of its approximations. **Tip**: you can precompute B for the highest k and take only some its vectors for each computation.

3. Spherical harmonics $Y_l^m(\theta, \phi)$ are the angular portion of the solution to Laplace's equation in spherical coordinates [1,2]. Compute and visualize the first few spherical harmonics (in spherical coordinates) on a sphere mesh. Compare your results to the basis B_i (eigenfunctions of W). Explain your results.

Submission

The HW can be done in either pairs or singles. Please submit a single **zip** file containing your report and MATLAB code using the electronic submission button in the course’s website. For evaluation of your work, we will mainly focus on the attached report, please consider it while composing your report and strive to make it as detailed as possible.

Good Luck!