Scientific Computing II

Assignment 2+

Due: 28/01/2019, 14:59

In this assignment we will implement the heat and the wave equation on an embedded triangle mesh. For this, we will employ first order finite elements. The required discrete Laplace operator is known as the cotan-Laplace.

For the assignment we will use the `trimesh` library[1] to simplify the handling of the triangle mesh as well as `pyglet`[2] for visualization. Python skeleton code demonstrating mesh loading and visualization is available here. Additional meshes are available (for example) in the CGAL repository.[3]

1.) Implementation of cotan-Laplace operator L. To obtain the expression for the matrix entries

$$L_{ij} = \left\langle \nabla\varphi_i, \nabla\varphi_j \right\rangle \tag{1}$$

of the cotan-Laplace we will proceed in a few steps. Please hand in the required calculations with your code and plots. (5 points)

a) Recall that linear finite elements on triangles are given by barycentric coordinates λ_i that is $\varphi_i = \lambda_i$. Compute the gradient $\nabla\varphi_i(x)$. In which direction is it pointing? Express the magnitude in geometric terms. (Note that it suffices to work with triangles in $\mathbb{R}^2$. It might also be helpful to first find the solution for the "standard" triangle with vertices $(0, 0)$, $(1, 0)$, $(1, 1)$.)

b) Compute the diagonal entries $L_{ii} = \left\langle \nabla\varphi_i, \nabla\varphi_i \right\rangle$. Express these as a function of the cotan of the triangles angles.

c) Compute the off-diagonal entries $L_{ij} = \left\langle \nabla\varphi_i, \nabla\varphi_j \right\rangle$. Express these as a function of the cotan of the triangles angles.

[1] https://pypi.org/project/trimesh/; the documentation is available here: https://trimsh.org/modules.html

[2] https://pypi.org/project/pyglet/

[3] https://github.com/CGAL/cgal/tree/master/Mesh_3/examples/Mesh_3/data

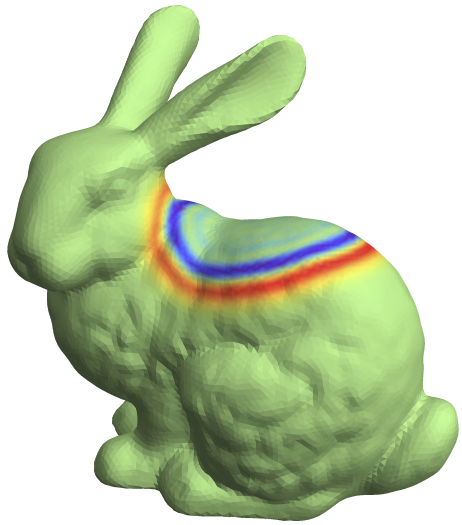

Figure 1: Solution of the wave equation on an embedded mesh.

d) Derive a compact formula for $(\Delta u)_i$, i.e. the cotan-Laplace at the i^{th} vertex.

e) Implement the cotan-Laplace. You should write a function that takes a `trimesh` triangle mesh object and returns the cotan-Laplace for it. The matrix should be stored as a sparse matrix (scipy provides one).

f) Implement the barycentric lumped mass matrix. It is given by

$$M_{ii} = \sum_j \frac{|T_j^i|}{3} \tag{2}$$

where j runs over all triangles T_j^i adjacent to vertex i and $|T_j^i|$ denotes a triangle's area. Write a function that returns the matrix for a given `trimesh` object.

g) To verify the operator, one can compute the mean curvature H of the mesh. In terms of the Laplace operator it is given by

$$\Delta x = 2\,H\,\vec{n} \tag{3}$$

where $x : U \subset \mathbb{R}^2 \to \mathbb{R}^3$ describes the surface locally (a so called embedding for it) and $\vec{n}$ the unit normal at the point. For the bunny, the mean curvature computed with your cotan-Laplace should match those shown in Fig. 2.

2.) Implementation of wave equation. (5 points)

a) Solve the wave equation

$$\frac{\partial^2 u}{\partial^2 t} - \Delta u = 0$$

with inital conditions being a Gaussian centered at a vertex (it should roughly be nonzero over the 1-ring neighborhood of the center vertex) using a suitable numerical scheme.

b) Generate visualizations of the solution after 1, 2, 3, 4, 5 seconds for different initial vertices.

c) Describe where in the plots numerical dispersion is visible.

3.) Implementation of heat equation. (5 points)

a) Solve the heat equation

$$\frac{\partial u}{\partial t} - \Delta u = 0$$

with inital conditions being a Kronecker delta over one vertex using a suitable numerical scheme.

b) Generate visualizations of the solution after 1, 2, 3, 4, 5 seconds for different initial vertices.

4.) *Bonus: Smoothing flows for surface meshes.* (3 points) When the vertex coordinates of the mesh are interpreted as function on it, then the heat flow can be used for mesh smoothing [1]. Demonstrate this for at least one suitable example (screenshots and a short description of your implementation are sufficient).

Please submit your implementation, notes and graphs before the deadline to wr@isg.cs.uni-magdeburg.de.

References

[1] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. "Implicit fairing of irregular meshes using diffusion and curvature flow". In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*. New York, New York, USA: ACM Press, 1999, pp. 317–324. URL: http://portal.acm.org/citation.cfm?doid=311535.311576.

Figure 2: Mean curvature H for the bunny mesh.