Mathematical Tools for Computer Graphics 2017

# Tutorial 13

The Poisson equation appears in various techniques in computer graphics. One of the best known ones is Poisson image editing.[1] In this tutorial we will solve the the Poisson equation in one dimension,

$$\frac{\partial^2 f}{\partial^2 x} = g(x), \tag{1}$$

using linear finite elements. As we showed in the lecture, this yields the matrix-vector equation,

$$L\,\bar{f} = M\bar{g} \tag{2}$$

where $L$ is the tri-diagonal stiffness matrix and $M$ the mass matrix, which is tri-diagonal as well; $\bar{f}$ and $\bar{g}$ are the basis function coefficient vectors for $f$ and $g$, respectively.

1.) Implement linear ("tent") finite element basis functions. Ensure that your implementation is reasonably efficient and in particular that it threads over lists.

2.) Implement signal reconstruction for finite element basis functions from a given coefficient vector.

3.) Implement the stiffness and mass matrix for Eq. 2 in Numpy.

4.) Solve the Poisson equation for:

- a constant right hand side;
- a linear right hand side;
- a random right hand side.

5.) Repeat the last experiment with $h$-refinement for different values of $h$.

[1] Pérez, Gangnet, and Blake, "Poisson image editing".

6.) Change your implementation to use SciPy's sparse matrix implementation and the SciPy's conjugate gradient solver `scipy.sparse.linalg.cg` that can exploit that $L$ is sparse, symmetric and positive definite. Compare the execution time as a function of $h$ for the dense and sparse versions.

## References

Pérez, P., M. Gangnet, and A. Blake. "Poisson image editing". In: *ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03*. Vol. 22. 3. New York, New York, USA: ACM Press, 2003, p. 313. URL: http://portal.acm.org/citation.cfm?doid=1201775.882269.