A Physically Based Mesh Generation Algorithm: Applications in Computational Medicine

David M. Weinstein Steven G. Parker Christopher R. Johnson
dweinste@cs.utah.edu sparker@cs.utah.edu crj@cs.utah.edu
Department of Computer Science
University of Utah, Salt Lake City, Utah

Abstract

The process of generating a finite element mesh is accelerated by integrating physical and geometrical constraints into the initial mesh. By beginning with a near-optimal initial mesh, fewer iterations of refinement are needed to obtain mesh convergence. We apply this algorithm to large-scale bioelectric field problems involving the complex geometries of the human body.

Introduction

One class of important applications in computational medicine is volume conductor problems which arise in electrocardiography and electroencephalography. To solve such problems, one first discretizes the physical geometry into elements, and then applies a finite element (FE) analysis to approximate the bioelectric fields throughout the discretized geometry [1]. Next, an error analysis is performed to find elements which need refinement. Additional elements are introduced and the process continues in an iterative fashion until some *a priori* convergence criteria is reached.

It is clear that mesh refinement would be unnecessary if one could somehow guess the final mesh from the start. In the face of complex geometries and inhomogeneities, this seems to be an impossible task. *However, if we could get close in the initial stage, fewer refinement steps would be required to reach the final stage.* For our grid to be nearly optimal, it must accurately reflect the geometry and the physics of our system. Effectively, it must be composed of small patches in the areas of high gradient and maintain the integrity of all boundaries.

To generate a grid with these properties, we "morph" or interpolate between the shapes of internal source boundaries and external insulating boundaries [2]. Morphing provides the geometric characteristics of the mesh, and the *rate* of the morph controls the density of the mesh. The intermediate interpolated shapes are resampled in space and "time" to provide the actual mesh points. For models with simple topologies, this method produces results similar to mapping methods for grid generation [3]. However, as we will demonstrate, the morphing method handles domains with complex topologies more gracefully.

Methods

As an initial testbed, we have constructed an algorithm to generate meshes from segmented MR images. Points lying upon heart and body contours were taken from a single transthoracic slice and used to define the sources and boundaries of the mesh. In two dimensions, the boundaries of the voltage sources, current sources, internal geometries, and the exterior of the model are closed represented by closed planar curves.

A curve in two-dimensional space may be represented in several different ways. These representations can be divided into three classes: parametric, explicit, and implicit [4]. For this application, we will exploit the properties of the implicit representation in order to perform shape interpolation.

The primary goal is to create an initial mesh which reflects areas of high gradients in the solution domain, while at the same time conforming to the topology of the physical domain. The boundary conditions and sources give rise to these gradients. For the human thorax model, the heart is considered to be a voltage source, and the body exterior an insulating boundary, while any additional defibrillator electrodes are modeled as current sources. We make the following assumptions about the different boundary conditions.

1. **Dirichlet**: Produce gradients which have a $\frac{1}{r}$ falloff, meaning that the size of the elements should be directly proportional to r. This would correspond to a voltage source in our problem domain, such as the endogenous fields of the heart.
2. **Neumann**: Specified by $\nabla\Phi \cdot \mathbf{n} = f$. For our purposes, the Neumann boundary conditions are used in two different ways.
   (a) $f = 0$: Used to specify an insulating boundary. In this case, the boundary has no effect on the final mesh other than shape. The body exterior is an insulating boundary.
   (b) $f = g(x, y)$: Used to specify a current source. In this case, the gradients exhibit a $\frac{1}{r^2}$ falloff from the boundary, which corresponds to elements of size proportional to r^2. Defibrillator electrodes are modeled as current sources.

Thus, the proper distribution for a voltage source (element size proportional to r), would require equally spaced contours, and the proper distribution for a current source (element size proportional to r^2), would require contours spaced proportionally to r.

Voltage source:

$$tD_{boundary}(x,y) + (1-t)D_{vsource}(x,y) = 0$$

Current source:

$$t^2 D_{boundary}(x,y) + (1-t^2)D_{isource}(x,y) = 0$$

where $D_{boundary}$, $D_{vsource}$ and $D_{isource}$ are the signed distance functions for the respective boundaries.

For a single problem which contains multiple sources of one type, the sources are combined into a single "meta-source" whose value is the minimum value obtained from each of the implicit equations.

If a single domain contains both voltage sources (Dirichlet boundaries), and current sources (Neumann boundaries), then the resultant curves must be combined. However, since the interpolation is done differently for the two cases, they cannot be combined into a single "meta-source". Instead, we must combine the two contours at each contour level. This is done to the implicit representations with the following function:

$$f(x,y) = |f_v(x,y) f_c(x,y)| \ \mathbf{Or}\ (f_v(x,y), f_c(x,y))$$

where the **Or** function is defined as:

$$\mathbf{Or}\ (a,b) = \begin{cases} -1 & \text{if } a < 0 \text{ or } b < 0 \\ 1 & \text{if } a >= 0 \text{ or } b >= 0 \end{cases}$$

This produces the union of the two curves, with the overlapping region removed.

The first step in building the mesh is to locate the isocontours. For each t value, the linear combination of the sources and boundary produces an implicit function for which we wish to trace the zero-set. Our trace algorithm finds a point on this set for each source, and traces along it until the contour is complete. The contour is then resampled to contain a fixed number of points. We repeat this process for each isocontour, and finally utilize a Delaunay triangulation method to mesh our points into an optimal configuration.

Results

Initial simulations show that the mesh generated using the morphing algorithm requires fewer iterations before converging. To obtain a relative accuracy of 5%, our algorithm required only 12 adaptation iterations, while a regularly sampled domain and a randomly sampled domain required 16 and 19 iterations, respectively. Furthermore, the final number of nodes in our fully adapted mesh is significantly lower than the number of nodes produced from the other initial meshes. Our fully adapted mesh contains 807 nodes, while the regularly sampled and the randomly sampled meshes contain 880 and 909 nodes.

Future Work

Ultimately, we will expand these tools to work for three-dimensional spaces. Most of the extensions are straightforward — isocontours become isosurfaces, and the same linear morphing still applies, but the surfaces are more difficult to parameterize. We would also like to use coherence obtained from the point generation algorithm in conjunction with an incremental Delaunay Triangulation algorithm to optimize the triangulation process.

Acknowledgments

The authors would like to thank Dr. John Schmidt for his valuable comments and suggestions. This research was supported in part by awards from the Whitaker Foundation and the National Institutes of Health.

References

[1] C.R. Johnson, R.S. MacLeod, and P.R. Ershler. A computer model for the study of electrical current flow in the human thorax. *Computers in Biology and Medicine*, 22(3):305–323, 1992.

[2] S.G. Parker, D.M. Weinsten, and C.R. Johnson. A morphing algorithm for generating near optimal grids: Applications in computational medicine. *University of Utah Technical Report*, UUCS-94-014, 1994.

[3] P.L. George. *Automatic Mesh Generation.* Wiley, New York, 1991.

[4] Brian Wyvill. Building models with implicit surfaces. *IEEE Potentials*, 1992.

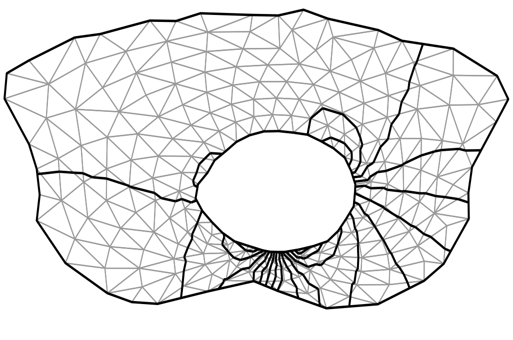

Figure 1: Heart and Torso meshed using Morphing algorithm (237 points)

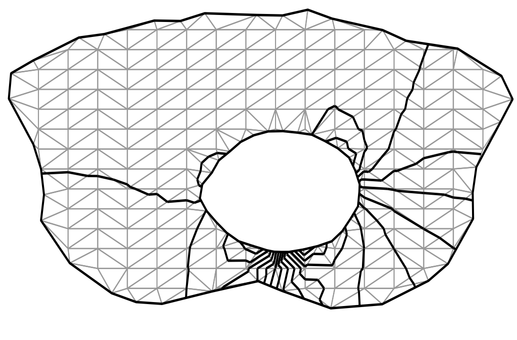

Figure 2: Heart and Torso meshed using regularly sampled points (237 points)