Department of Scientific Computing
Florida State University

Computational Exposition 2009

Spherical Centroidal Voronoi Tessellations Unstructured Meshes For Ocean Applications

Geoff Womeldorff

Collaborators: Max Gunzburger & Janet Peterson

Abstract : In the past and increasingly so towards the present, the modeling of our oceans is of great import. In order to facilitate this simulation, we wish to discretize the sphere in an optimal manner. That is to say, we wish to place the most grid points where we have the most information to capture. Here, we wish to support both local refinement and global refinement. In order to do this, we use the theory of centroidal Voronoi tessellations on the sphere to give us a regular mesh, and then we add to it the ability to refine locally by managing the density of generators through a custom density map. We also take into account an approximation of the shoreline and we use this to force grid cells to conform to it.

What is a Spherical Centroidal Voronoi Tessellation?

A Spherical Centroidal Voronoi Tessellation (SCVT) is a CVT where the domain in question is a sphere. To be precise, let S be a 2-sphere and $\{z_i\}_{i=1}^n$ be a set of points, called *generators*, on S. Define subsets of S, denoted V_i and called *Voronoi regions* , where

$$V_i = \{x \in S | \; |x - z_i| < |x - z_j| \text{ for } j = 1, \ldots, n, j \neq i\} \tag{1}$$

and $|\cdot|$ is the Euclidean norm in $\mathbb{R}^3$. In addition, we ensure that the set of V_i are a *tessellation* by prescribing that

$$V_i \cap V_j = \emptyset, \; i \neq j \tag{2}$$

$$\cup_{i=1}^n \overline{V_i} = \overline{S} \tag{3}$$

The final property we specify is that $z_i = z_i^*$,

$$z_i^* = \frac{\int_{V_i} y\rho(y)dy}{\int_{V_i} \rho(y)dy} \tag{4}$$

and so the generator of each V_i is its mass center z_i^*. A set of V_i's satisfying (2) and (3) is a *tessellation* , adding (1) makes it a *Voronoi tessellation* , and appending (4) creates a *centroidal Voronoi tessellation* - which we make into an SCVT by simply saying that S is a sphere.

How do we create an SCVT?

Typically we create an SCVT through an iterative process known as Lloyd's method; from S. Lloyd of Bell Laboratories in the 1960's. Simply, we take our set of generators, create a Voronoi diagram, then move each generator to the mass center of its region. We do this until some stopping criteria have been met, typically choosing a maximum number of iterations, as a safeguard, and the second criterion as the maximum change between two iterations of a particular generator as being less than some epsilon.

Here is Lloyd's algorithm, more formally:

Given:

- S a 2-sphere
- n a positive integer
- $\{z_i\}_{i=1}^n$ an initial set of generators
- ρ a density function defined on S

Iteration:

1. Create a Voronoi tessellation using $\{z_i\}$.
2. Calculate the mass center of each V_i.
3. Set the generator of each V_i to its mass center.
4. Repeat 1 - 3 until the convergence criterion (or criteria) have been satisfied.

One note however, in practice we use STRIPACK to compute the Delaunay Triangulation on the sphere, and compute the Voronoi diagram from this triangulation. We do not directly compute the Voronoi diagram, but the spirit of the algorithm above remains the same.

Density Functions via Images

Traditionally we control placement of generators in an (S)CVT via an analytic density function. As an example, we could define a piecewise function taking the z coordinate as input to produce one density in an equatorial band, and another, smaller, density on the rest of the sphere. Suppose, though, that we might wish to create an SCVT that would be used as a mesh for a PDE simulation of climate over the earth. We may desire that the generators are more dense in the water than over land. This is certainly not something we could do with an analytic density function; that is, capturing coastlines in any sort of accurate manner. So, in order to describe an intuitive approximation of the density of Voronoi regions over the sphere we suggest the use of an image, a set of RGB values, where we use the red channel as a density value from 0 to 1. This will allow us to set, proportionately, the amount of generators that we desire to have in regions on the sphere; to increase their density by the shore, and decrease their density farther away - for example.

Density Function via Image Algorithm:
Let x be a point on the unit sphere in Cartesian coordinates, s a point on the unit sphere in spherical coordinates, and v a point in an image of dimensions (w, h).

1. Transform x into s
   (a) Calculate the radial coordinate of s, $s_1 = \sqrt{x_1^2 + x_2^2 + x_3^2}$.
   (b) Calculate the azimuthal coordinate of s, $s_2 = \arctan(x_2/x_1)$.
   (c) Calculate the polar coordinate of s, $s_3 = \arccos(x_3/s_1)$.
2. Transform s into v
   (a) Calculate the first coordinate of v by mapping s_2 from the range $(-\pi, \pi)$ into the range $(1, w)$.
   (b) Calculate the second coordinate of v by mapping s_3 from the range $(0, \pi)$ into the range $(1, h)$.
3. Select the pixel at the location v in the image p and return the value of the red channel at this pixel.

Example Density Image and Resulting SCVT

Here we present a density image wherein we prescribe that the generators should be most dense in the region surrounding the shoreline, both inside and outside of the shoreline (represented as the color blue).

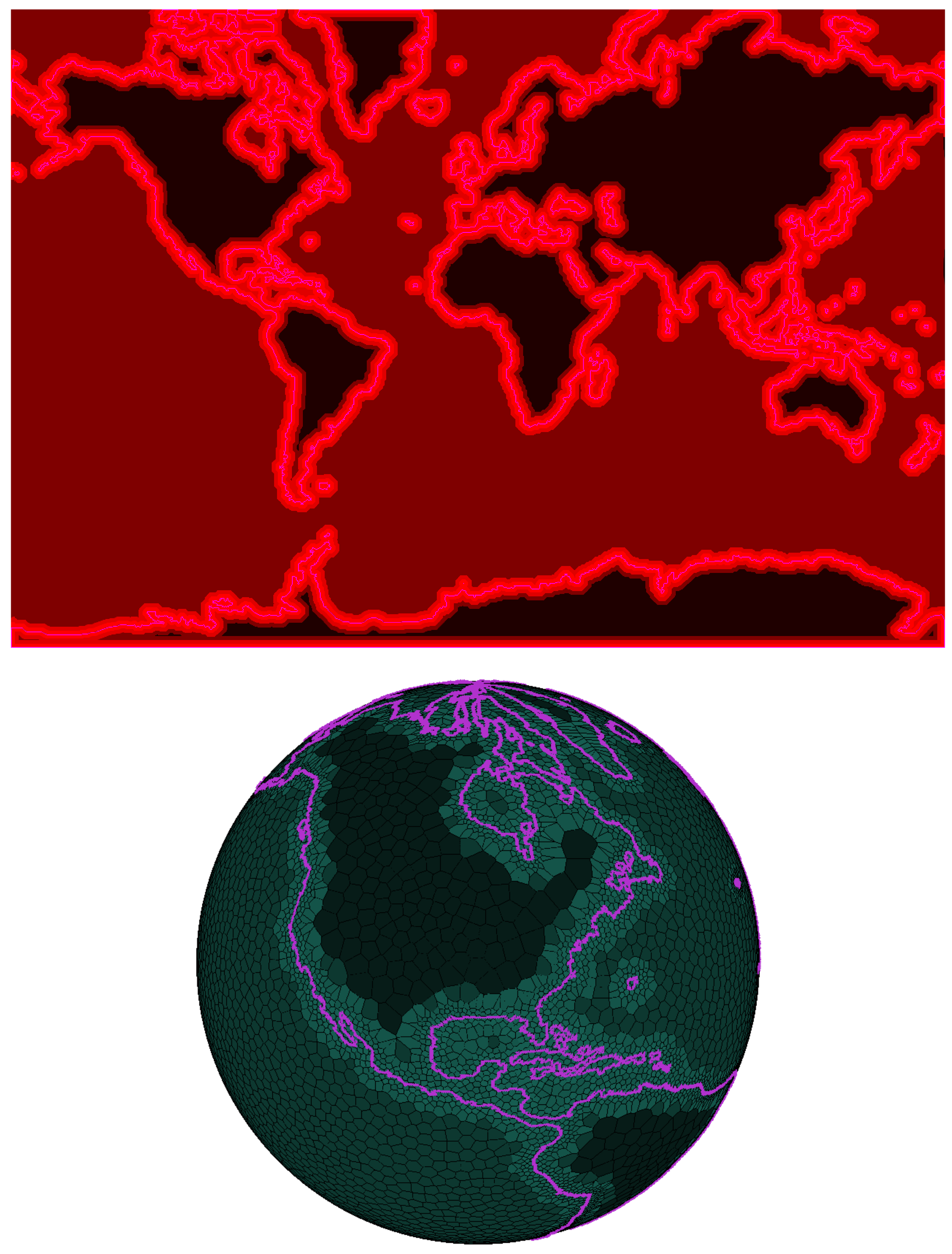

Figure 1: Top: example density image, Bottom: a portion of the resultant SCVT mesh

Shoreline Conforming SCVTs

We now describe a method to more accurately capture the shoreline that is compatible with the Lloyd iterations we are using to produce our SCVTs. We include two additional steps in Lloyd's Algorithm between Steps 3 and 4. These extra steps will help to move some Voronoi generators into a set of potential positions that we define, allowing us to have, much more precisely, a mesh which approximates the actual shorelines of the earth. This is of particular interest in that through this we can facilitate the communication of models which operate over the land and those that operate over the ocean by allowing them to use subsets of the same mesh. For our purposes here, the term 'shoreline point' means one value in an array which describes an approximation of the actual shoreline.

Shoreline Conforming Algorithm:

1. For each point on the shoreline, associate with the shoreline point the generator that is closest, recording the distance.
2. For each generator that is associated with a point on the shoreline, move the generator's location to be coincident with the shoreline point which is the closest of those so associated.

Example of Shoreline Conforming Algorithm

Here we present several sample images which are of the same portion of an SCVT mesh which was created by, in the first case, having the shoreline algorithm turned off, and in the second case by having the shoreline algorithm turned on. Note that moving the Voronoi generators (triangulation vertices) to the shoreline points causes the triangulation to conform to the prescribed set of shoreline points.

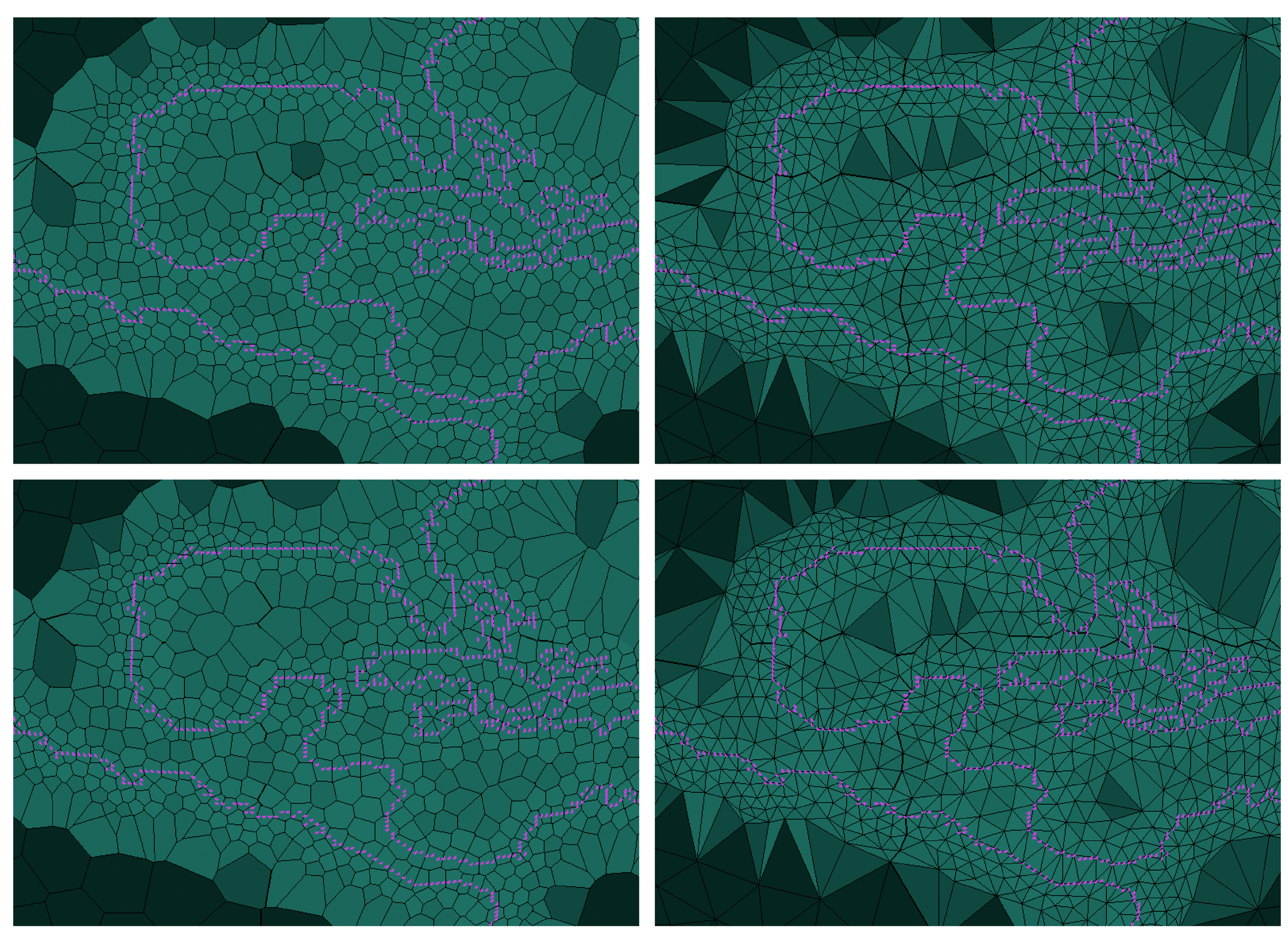

Figure 2: Top Row, left to right: Shoreline Off, Voronoi mesh and triangulation. Bottom Row, left to right: Shoreline On, Voronoi mesh and triangulation.