Department of Scientific Computing
Florida State University

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. To start, we use the theory of centroidal Voronoi tessellations on the sphere to give us a regular mesh. We supplement this with an approximation of the shoreline, which we use by both forcing generators to conform to it and regulating the density of generators based on their distance to it. In addition, away from the shoreline we vary the density based on some proxy information.

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.

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. We provide an example set of shoreline points in Figure 1.

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 associate

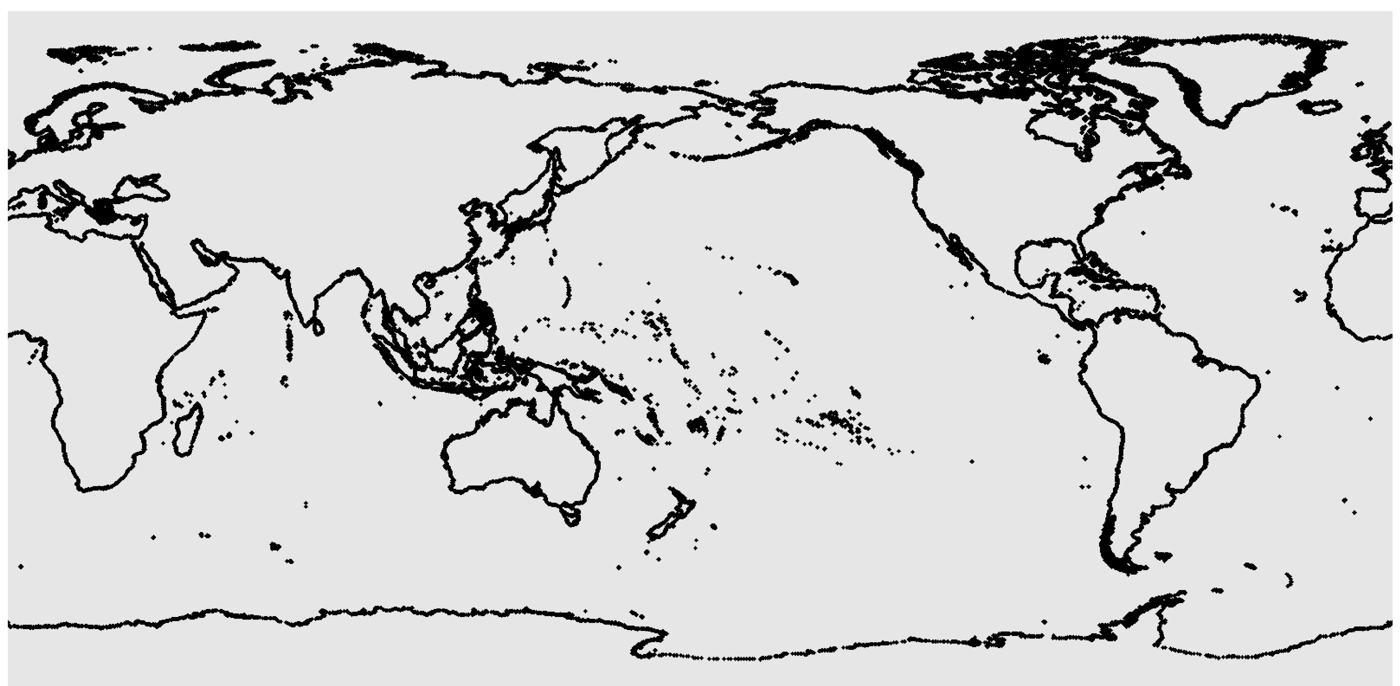

Figure 1: Longitude/latitude coordinates (black) in the Intermediate (1.0km) version of the GSHHS database

Density Based on Distance to Shoreline

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. We may also desire to increase the density (thus reduce the size) of the generators nearer to the shoreline as opposed to those further from the shoreline in the body of the oceans. 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. Each iteration in our grid generation we calculate the distance to the closest shoreline point for each Voronoi generator and triangle circumcenter. We then calculate a distance, $dist_\alpha$, by consulting this list of distances, which for generators/circumcenters having a distance greater than $dist_\alpha$ we will not influence the density based on distance. After this $dist_\alpha$ cutoff region we will only use some position dependent proxy information (kinetic energy, surface height, for example). Where $dist < dist_\alpha$, we will let both proxy and distance compete to influence the density.

Density Based on Distance to Shoreline Algorithm:

Let x be a point on the unit sphere in Cartesian coordinates, let $proxy(x)$ be the value of the density proxy at x, let $dist(x)$ be the arc distance to the closest shoreline point from x, let $dist_{max}$ be the largest distance to a closest shoreline point from all of Voronoi generators and triangle circumcenters for the current iteration, let $dist_\alpha$ be the grid dependent cutoff point for density influenced by distance to shoreline, and let β be a grid scaling parameter:

$$\rho(x) = \begin{cases} \max\left(\tanh\left(\frac{1-dist(x)}{dist_\alpha}\right), proxy(x)\right)^\beta & : x|dist(x) \leq dist_\alpha \\ proxy(x)^\beta & : x|dist(x) > dist_\alpha \end{cases} \tag{5}$$

Example Meshes

In Figures 2 and 3, we show a mesh composed of 100,000 generators. Here, the shoreline is represented as a set of magenta points, and the coloring of the Voronoi cells are relative to their density value (high = bright, ..., low = dark). Soon, we plan to support multiple proxies in different continents / land masses. This is of particular interest in that through this we can facilitate the communication of models which operate over various regions of the earth (land, sea, land ice, sea ice) by allowing them to use subsets of the same mesh.

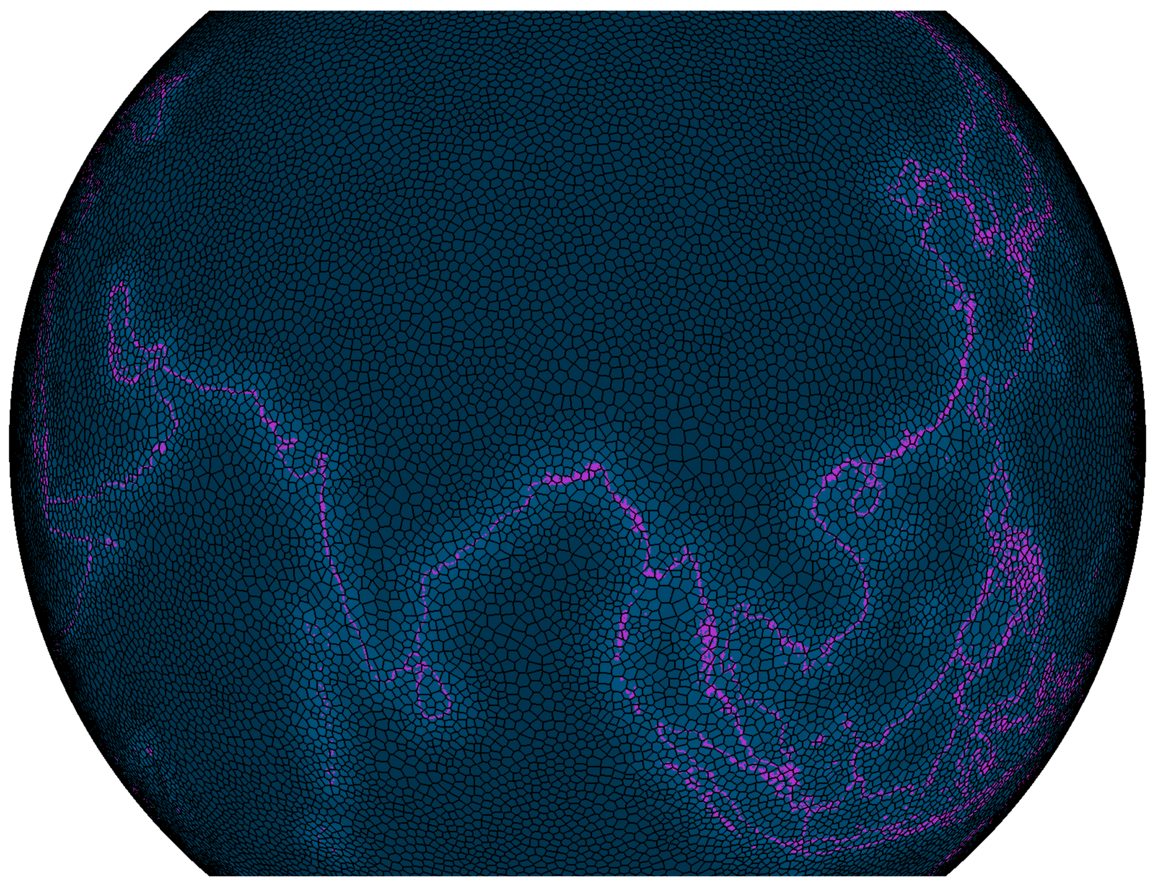

Figure 2: View of Asia, 100k generators, β 4.0

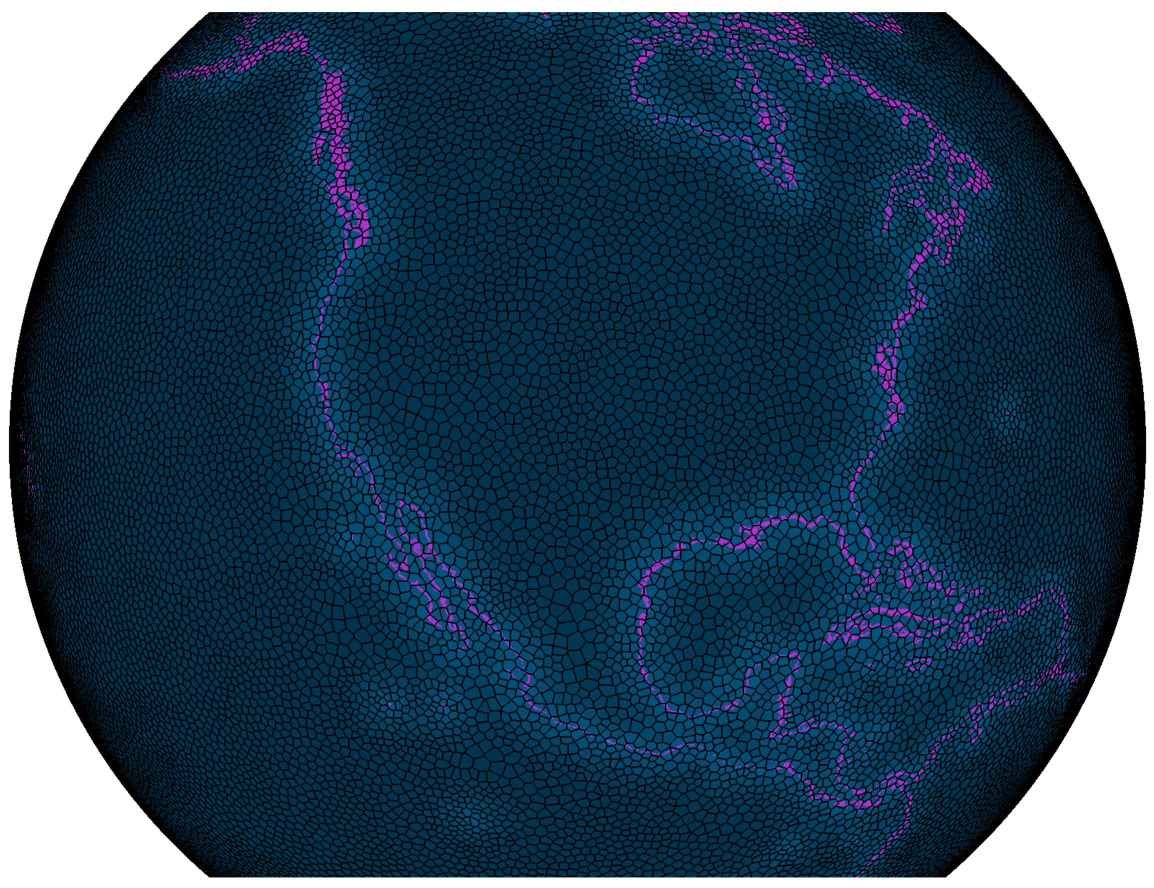

Figure 3: View of North America, 100k generators, β 4.0

References

[1] Q. Du, V. Faber, and M. Gunzburger, *Centroidal Voronoi Tessellations: Applications and Algorithms*, SIAM Review, Vol. 41, No. 4, 1999

[2] P. Wessel and W. Smith, *A global, self-consistent, hierarchical, high-resolution shoreline database*, Journal of Geophysical Research, Vol. 101, No. 4, 1996

[3] T. Ringler, L. Ju, and M. Gunzburger, *A multiresolution method for climate system modeling: application of spherical centroidal Voronoi tessellations*, Ocean Dynamics, Vol. 58, 2008