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 be a 2-sphere and \( \{s_i\}_{i=1}^n \) be a set of points, called generators, on \( S \). Define subsets of \( S \), denoted \( V_i \) and called Voronoi regions, where

\[
V_i = \{ x \in S | |x - s_j| < |x - s_i| \text{ for } j = 1, \ldots, n, j \neq i \}
\]

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

\[
V_i \cap V_j = \emptyset, \ i \neq j
\]

\[
\sum_{i=1}^n |V_i| = S
\]

The final property we specify is that \( ||\cdot|| \) and a 2-sphere and a toroidal Voronoi Tessellation (SCVT) is a CVT where the domain in question is a sphere. To be precise, let

Let \( \rho(x) \) be a positive density function defined on \( S \). A set of \( \{s_i\}_{i=1}^n \) \( n \) an initial set of generators

\[
\rho \text{ a density function defined on } S
\]

and \( \{s_i\}_{i=1}^n \) an initial set of generators

\[
\rho(x) = \frac{1}{n} \int_S \rho(y) \, dy
\]

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
- \( \{s_i\}_{i=1}^n \) an initial set of generators
- \( \rho \) a density function defined on \( S \)

**Iteration:**
1. Create a Voronoi tessellation using \( \{s_i\} \).
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 SCVT 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 region surrounding the shoreline, both inside and outside of the shoreline (represented as the color blue).

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.

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.