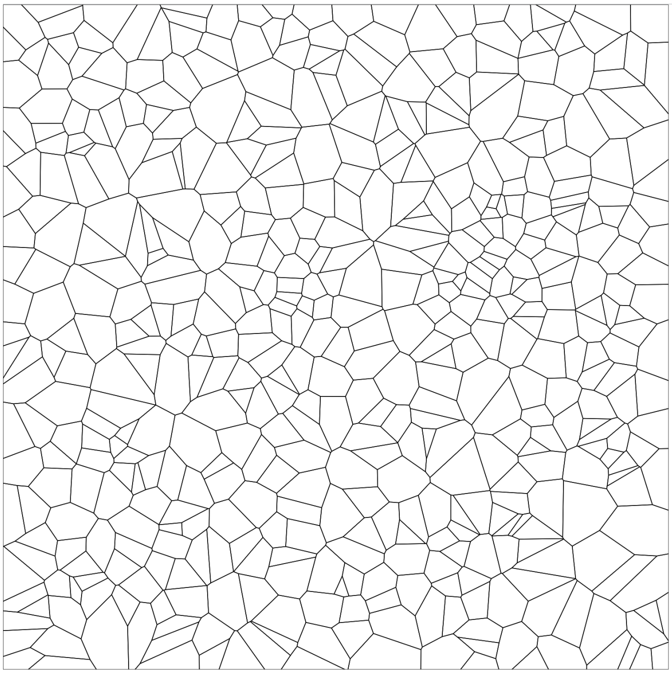
CSE306 Assignment 2 Report

Note: All the images in this report can be found in the folder Assignment1 under the name underlined.

I first implemented the Sutherland-Hodgman polygon clipping algorithm. To do so, I created the classes <Vector>, <Edge> and <Polygon> and then added the function clipPolygonfunction() following the pseudo-code in Section 4.2. Here is an example of the output we obtain when the **clipPolygon** is created using **{Vector(0., 0.), Vector(0., 1.), Vector(1., 1.), Vector(1., 0.)}** and the **subjectPolygon** is created using **{Vector(1., 0.), Vector(1, 0.5), Vector(0.5, 0.7)}**

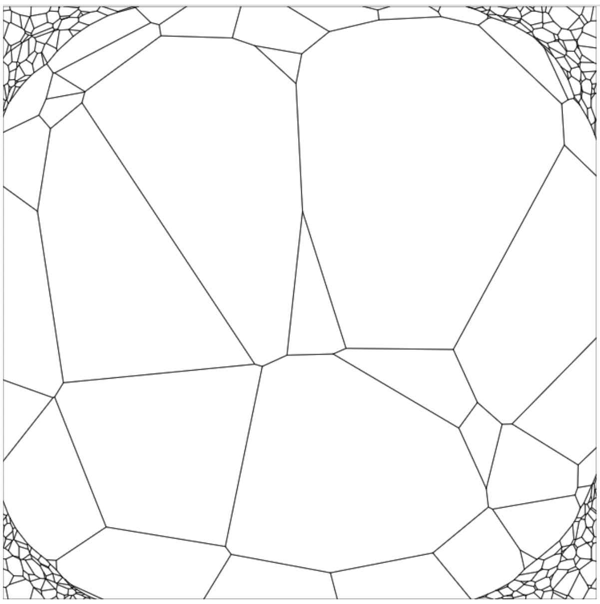
clip.svg: Clipping the polygon with the Sutherland-Hodgman algorithm

Then I implemented the Voronoï Parallel Linear Enumeration algorithm in 2D. I added the voronoi() function in my *clippolygon.cpp* file and modified my main function. This is the result we obtained:

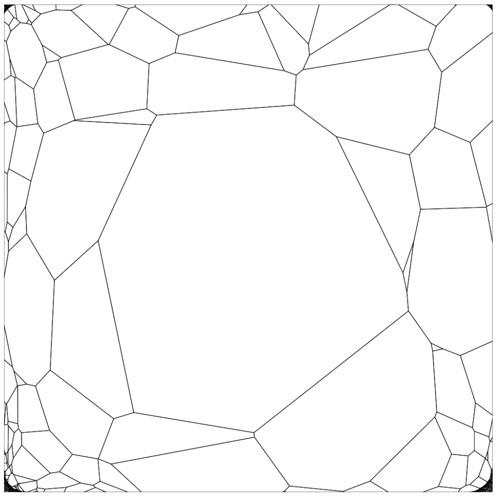


Voronoi.svg: Voronoi diagram with a subject polygon with n=500 random vertices

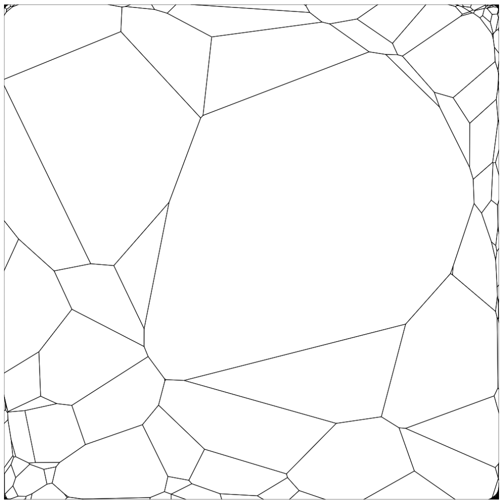
I first implemented a gradient descent method for semi-discrete optimal transport. I saved a svg file under the name *opti_k.svg* every 50 iterations. However this algorithm converges slowly. This is shown for example by the following outputs:



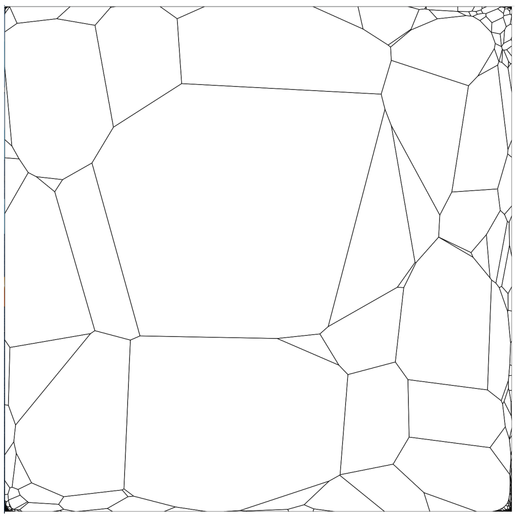
opti_50: Power-diagram optimized using gradient descent after 50 iterations



opti_650: Power-diagram optimized using gradient descent after 650 iterations



opti_1600: Power-diagram optimized using gradient descent after 1600 iterations



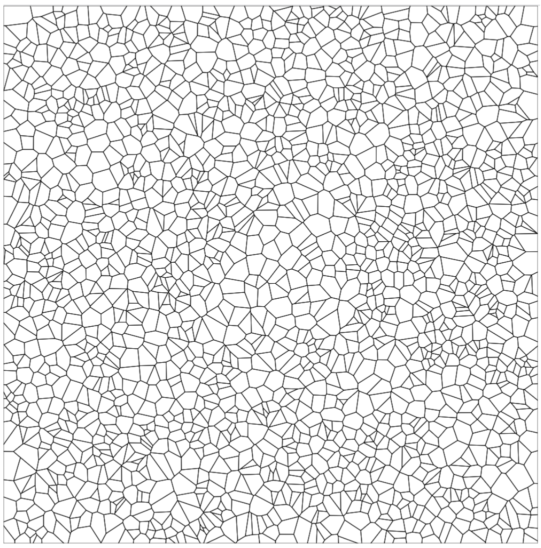
opti_2600: Power-diagram optimized using gradient descent after 2600 iterations

Therefore, as the previous method was slow, I implemented the semi-discrete optimal transport in 2d using L-BFGS. I first added the power diagram functionality.

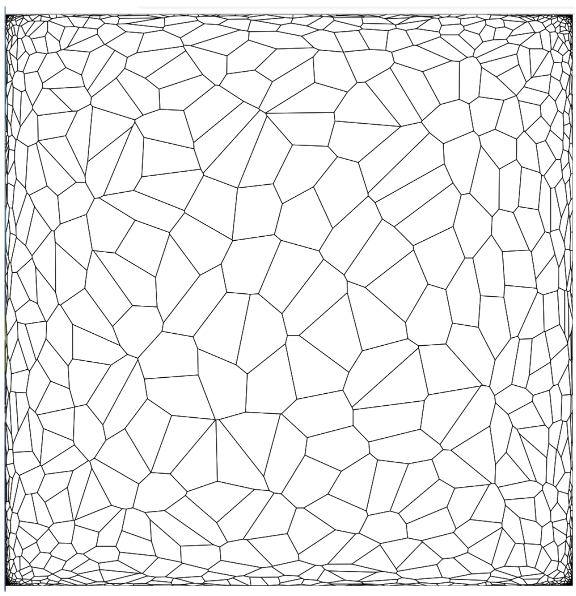
Then I implemented the evaluate function with the weights as variables passed in parameter. I used a density such that so that the cell associated to a site at position yi has an area proportional to

$$\exp(-\|y_i - C\|^2/0.02)$$

where C is the center of this unit square. Before optimisation the Voronoi diagram is:



beforeOptimisation: Power diagram



afterOptimisation: Power-diagram optimized using semi-discrete optimal transport using L-BFGS after 1400 iterations