Assigment 2

Atodiresei Matei

June 16, 2024

1 Introduction

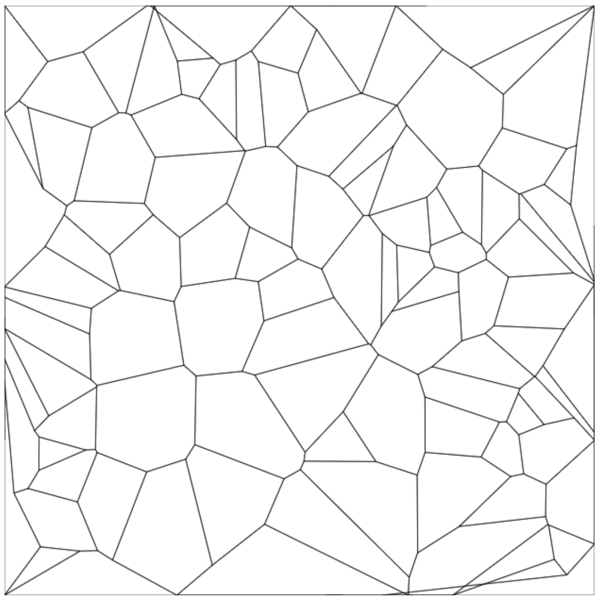
The final goal of this project was to create a fluid simulator. This required these three steps: built a basic geometry processor that can create voronoi diagrams, expand this geometry processor by implementing a semi-discrete optimal transport and then finalize with computational fluid dynamics.

2 Voronoi Diagrams and Clipping

I started off by creating the basic polygon class, same as in the previous assignment, but in only 2 dimensions. Then we implemented the Sutherland-Hodgman polygon clipping algorithm.

To create a Voronoi Diagram we use the template from the lecture notes as well as the functions inside_voronoi and intersect. The first function checks if a point is inside the cell of another(based on formula : $\langle X-M, P_j-P_i\rangle < 0$) and the second one computes where the bisector of 2 points intersect the line created by other 2 points. $P = A + t(B - A)$ where $t = \frac{\langle M-A,P_i-P_j\rangle}{\langle B-A,P_i-P_j\rangle}$

If we initializing some points we can use this algorithm to create the Voronoi Diagram by looping over all the points and adding the created polygons.


(a) A Voronoi diagram with 100 points generated in 0.01059s(no kd-tree is implented

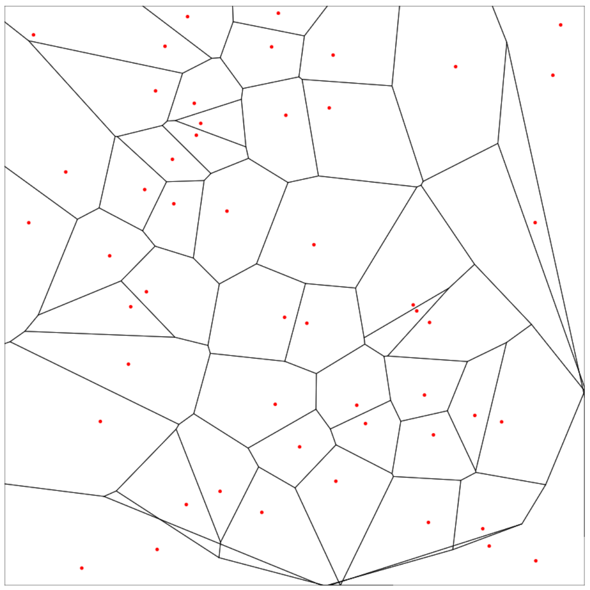

(b) A Power Diagram with 50 points generated in 0.00377s(no kd-tree implented)

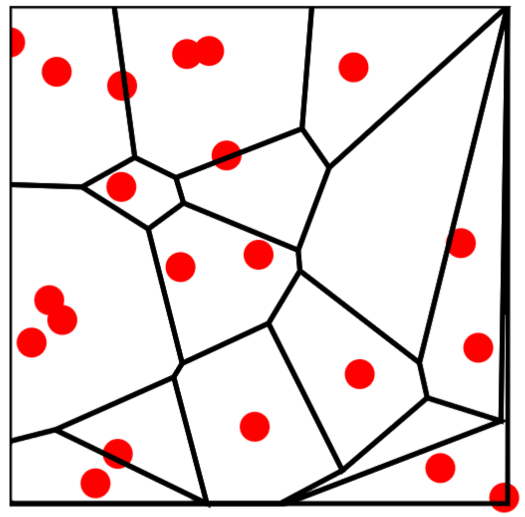
Figure 1: Caption for both images

3 Power Diagrams and Semi-Discrete Optimal Transport

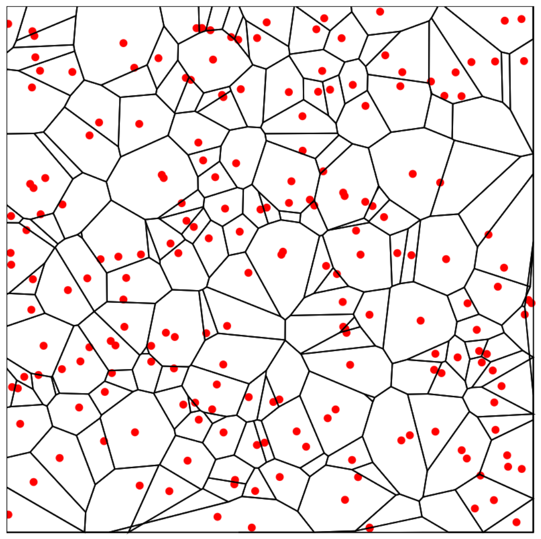
Then I implemented power diagrams. This is just a simple class that is used for storing a "state" with all the weights and positions of points and polygons.

When implementing power diagrams I also had to change some of the functions so that they are influenced by the weights. These functions where inside_power and intersect_power. Their formulas were changed to: $\|P-P_i\|^2-w_i \leq \|P-P_j\|^2-w_j$ for the inside or outside formula and $M' = M + \frac{w_i-w_j}{2\|P_i-P_j\|^2}(P_j - P_i)$.

These formulas seem to perform basically as expected.



(a) A Power Diagram where it clear that the weight have effect(the weights are 1 for most of the points, but for 1/5 of them it is 100)



(b) A Similar Power Diagram, but with many more points

Figure 2: Caption for both images

However when I tried implementing a custom set of weights, for example: centered Gaussian. There seemed to be no difference as seen in this image 3

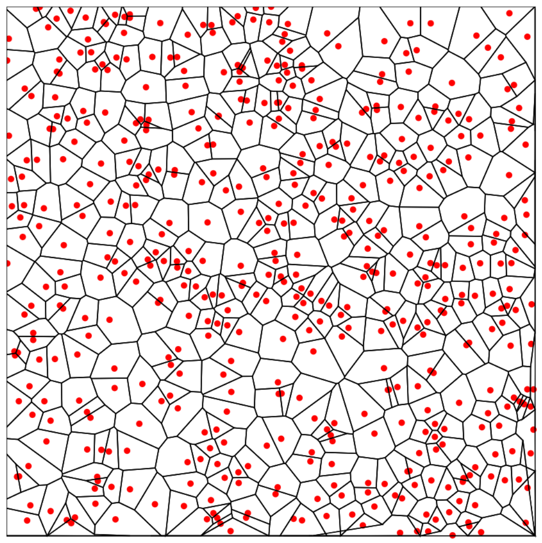


Figure 3: A Power Diagram, that should have weights based on a centered Gaussian

In this class we also implement some functions that are used for the optimization using L-BFGS.

I tried following the sample code and implementing static lbfgsfloatval_t _evaluate, lbfgsfloatval_t evaluate, static int _progress and int progress, however the optimization did not work in the end.