CSE306: Fluid Solver Report

MAMOUN ELKHETTAR

June 15, 2020

This document reports the step taken and the work achieved to build a free-surface 2D fluid solver using in-compressible Euler's equations in $C{+}{+}$ for the class CSE306.

1 Voronoï Parallel Linear Enumeration

In this part, we want to build a Voronoï diagram using the Voronoï Parallel Linear Enumeration. To do so, we first implement the Sutherland-Hodgman polygon clipping algorithm (in **polygon.cpp**). We create a **Polygon** class which has a **std::vector** of **Vectors** as vertices. We follow the algorithm in the lecture notes and we write to functions **intersect** and **inside**, to compute the point of intersection between the (finite) edge [preVertex, curVertex] and the (infinite) line.

After that, we modify the Sutherland-Hodgman polygon clipping algorithm to get the clipping using the Voronoï Parallel Linear Enumeration. The output is now a vector of Polygon that we construct iterating on the input **Polygon**. We also write a new $save\_svg\_voronoi$ function to handle the print of vectors. This change for the $save\_svg$ function was indicated to me by Carolina.

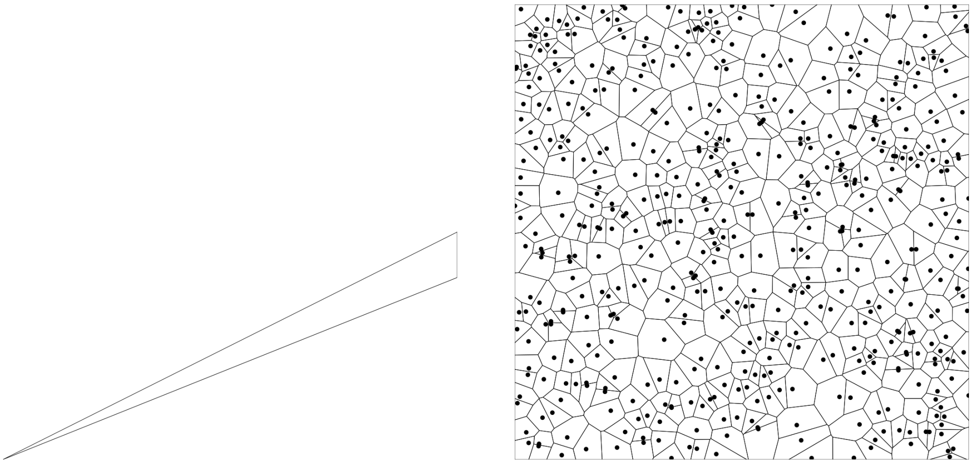
We can now get Voronoï diagrams with the number of points we want. Figure 1 display the results.

2 Power diagram and weights optimization with LBFGS

Here again, we modify the intersection and inside function to compute the weights and the new norm. For the power diagram we get Figure 2.

3 Lab 5 and 9

I have tried both lab 5 and 9. Lab 5 is almost fine where as lab 9 is still not compiling correctly.



(a) Polygon with Sutherland-Hodgman polygon clipping (0.3 seconds to compile)

(b) Voronoï diagram with 500 points (25 sec to compile)

Figure 1: Voronoï Parallel Linear Enumeration

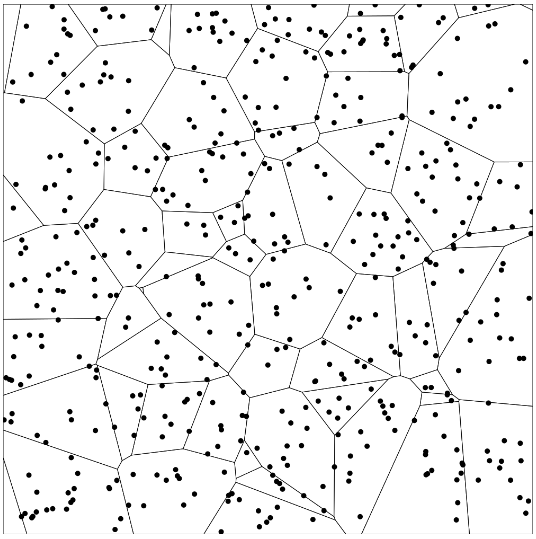


Figure 2: Power diagram without optimization