Howe Work 02

due 2018-12-11
Please send pdf of your homework to `si@wias-berlin.de`
please include your student ID in the title of your email.

Problem 1 (4 credit) Minimum spanning tree

Given a set of n points in the plane, we can think of the points as defining a Euclidean graph whose edges are all $\binom{n}{2}$ (undirected) pairs of distinct points, and edge (p_i, p_j) has weight equal to the Euclidean distance from p_i to p_j . A minimum spanning tree is a set of $n-1$ edges that connect the points (into a free tree) such that the total weight of edges is minimized.

- Prove that all edges of a minimum spanning tree of a graph $G = (V, E)$ belong to the Delaunay triangulation of its vertex set V. (2 credit)
- Show a minimum spanning tree of a finite point set of 10 points. (Hint: you could use Detri2 to create a set of points, also to dump the pictures into a .png file) (2 credit)

Problem 2 (8 credit): Growing empty circumcircles

We describe an approach which was used by Delaunay himself. It is a process of "growing empty balls" within a set of points. It starts with an empty ball at any point $\mathbf{p} \in S$, see Figure 1 (1). Let this circle grow as long as it does not touch any other points of S. This process stops once it touches a point $\mathbf{q} \in S$, we get a Delaunay edge $\mathbf{pq}$ of S, see Figure 1 (3). Now we grow an empty circumcircle of $\mathbf{pq}$ by choosing one of the two possible directions along the bisector line of $\mathbf{pq}$, see Figure 1 (4). This process stops in one of the two cases: it either (i) touches a third point $\mathbf{r} \in S$, or (ii) never touches any point of S. In case (i), we find a Delaunay triangle $\mathbf{pqr}$. While in case (ii) the edge $\mathbf{pq}$ must be a convex hull edge of S, and we simply switch the

search direction to the opposite. This process will continue as long as there are vertices of S which do not belong to any Delaunay triangle.

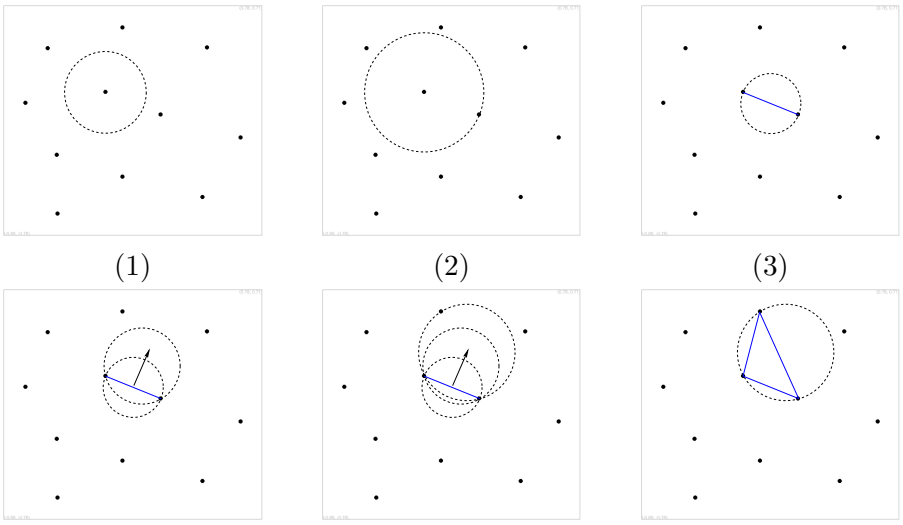

Figure 1: Growing empty circles to get Delaunay simplices.

(1) Given a set of n points in the plane, what is the worst-case runtime of this approach.(2 credit)

(2) Write a program which implements this approach. You could use any programming language, for example, C, C++, Matlab, etc. The program should be able to read a set of points, and output a set of Delaunay triangles. (6 credit)

Hint: In task (2), you could use Detri2's file formats

- `.node` file, a list of 2d points,
  see `https://www.cs.cmu.edu/~quake/triangle.node.html`, and
- `.ele` file, a list of triangles,
  see `https://www.cs.cmu.edu/\~quake/triangle.ele.html`

so that you could use Detri2 to create an arbitrary point set, and visualise the resulting Delaunay triangles as well.