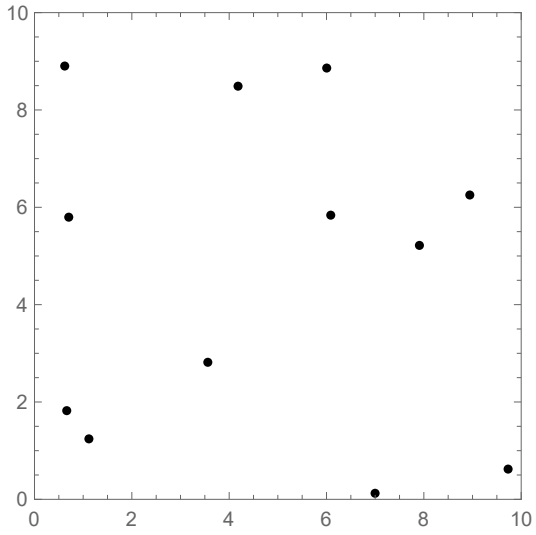
Triangulation Algorithms

MATH 261 Computational Geometry

Consider the following algorithm for triangulating a set of points S in the plane.

> Triangle-Splitting Algorithm: Find the convex hull of S and triangulate this hull as a polygon. Choose an interior point and draw edges to the three vertices of the triangle that contains it. Continue this process until all interior points are exhausted.

1. Use the triangle-splitting algorithm to triangulate the following point set S:

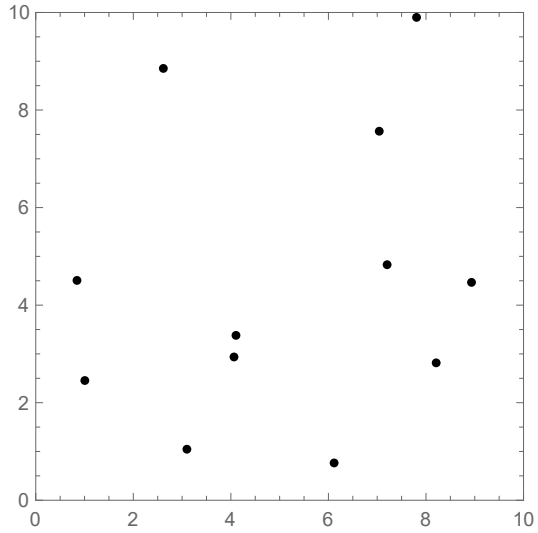


2. Depending on how you triangulate the convex hull and the order in which you choose the interior points, can the triangle-splitting algorithm produce *all* possible triangulations of S?

3. What is the computational complexity of the triangle-splitting algorithm?

Incremental Algorithm: Sort the points of S by their x-coordinates. The first three points determine a triangle. Consider the next point p in the ordered set and connect it with all previously considered points $\{p_1, \ldots, p_k\}$ which are visible to p. Continue this process of adding one point of S at a time until all of S has been processed.

4. Use the incremental algorithm to triangulate the following point set S:



5. If you are allowed to rotate the set of points arbitrarily before applying the incremental algorithm, does the algorithm produce *all* possible triangulations of S?

6. What is the computational complexity of the incremental algorithm?