Computational topology
Lab work, 4th week

Line-sweep triangulation of a point cloud

Suppose we are given a set of points

$$P = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

in the plane $\mathbb{R}^2$ (represented in their cartesian coordinates).

Our task is to construct a triangulation of the convex hull of P, i.e. of the polygon $\mathrm{Conv}(P)$, via vertical line sweep.

The geometric intuition behind the algorithm is simple:

- Start with an empty set K (this will eventually become our 2-dimensional simplicial complex determining the triangulation).
- Imagine a vertical line in the plane sweeping from left to right.
- When the line hits a point $T \in P$:
  - add that point to K,
  - for every vertex to the left of T add an edge from T ending in that vertex if and only if that vertex is *visible* from T, and
  - for every triangle formed by the vertex T and an edge already in K add the triangle to K if and only if that edge is visible from T.

We must now make precise the meaning of 'visible'. Suppose that K has been partially constructed and the vertical line has hit the next point $T \in P$ (which is not yet a vertex of K). Call a vertex V of K *visible from* T if the segment TV does not intersect any edge of K (apart from the two edges containing V). Similarly, call an edge $E = V_1V_2 \in K$ *visible from* T if the triangle TV_1V_2 does not intersect any other edge from K. Note that the edge $E = V_1V_2$ is visible from T if and only if the midpoint of that edge is visible from T, so we only need to explain how to check for visibility of single points (vertices or edge midpoints) from T.

1 Intersection of two planar segments

Given two segments AB and CD in the plane $\mathbb{R}^2$ we would like to determine if these segments intersect. Each segment PQ is represented by its boundary points $P(x_P, y_P)$ and $Q(x_Q, y_Q)$ or, equivalently, by the corresponding spatial (column) vectors $\mathbf{r}_P$ and $\mathbf{r}_Q$.

1. Parametrize the line through A and B and the line through C and D. Write down the system of linear equations, which can be used to determine the point of intersection of the two lines. Find the additional conditions, which will assure that the segments AB and CD actually intersect.
2. Write a Julia function `do_segments_intersect` which returns `true` if the segments intersect and returns `false` if they don't intersect.

2 Actual line-sweep function

We will use the function `do_segments_intersect` to test for vertices from K which are visible from T (a point which was just hit by the sweeping vertical line). Moreover, note that only boundary vertices of $|K|$ can be visible from T, so we will keep track of these boundary vertices at each step. (Note that the set of boundary vertices is, at least usually, much smaller than the set of all vertices.)

We do assume that the points in P are in general position, so no two points share the same x-coordinate and no 3 points lie on the same line.

Write a Julia function `line_sweep` which returns the line sweep triangulation from a given set of points. Use the pseudocode below as an aid, but be cautious when following it. Additional checks are necessary to avoid unwanted intersections, e.g. the segment TV will intersect the segment VW (in the vertex V), but that does not yet mean that V is not visible from T. Also, a certain amount of bookkeeping is necessary to keep track of the boundary of the currently constructed triangulation.

```
Sort points in $P$ by increasing $x$-coordinate.
Construct the triangulation of the first 3 points in $P$. (The first triangle is needed to
  start.) Store the current constructed triangulation in $K$ and the boundary vertices
  (i.e., the first three points from $P$) in $B$.
for $T \in P[4:\text{end}]$ do
    Construct two lists, visible_v and visible_e, of true boolean values with the same
      length as $B$.
    for $V \in B$ do
        for edge $V_i V_{i+1}$ of two consecutive vertices from $B$ do
            if segments $TV$ and $V_i V_{i+1}$ intersect then
                visible_v[index of $V$] &= false
            end
            // Let's denote by $M$ the midpoint of the segment from $V$ to the
               next vertex in $B$.
            if segments $TM$ and $V_i V_{i+1}$ intersect then
                visible_e[index of $V$] &= false
            end
        end
    end
    Add vertex $T$ to $K$.
    Add edges $TV$ to $K$ for $V \in B$ corresponding to true values in visible_v.
    Add triangles $TV_iV_{i+1}$ for consecutive vertices $V_i, V_{i+1} \in B$ corresponding to true
      values in visible_e.
    Remove vertices corresponding to true values in visible_v from $B$ (apart from the
      two 'boundary' ones).
    Add vertex $T$ to $B$ in place of the removed vertices.
end
```

3 Improvements and alternative approaches

1. Note that a boundary vertex $V \in B$ is visible if and only if it is a vertex of a visible boundary edge. Hence, keeping track of both, visible boundary vertices and visible boundary edges, is redundant. Improve the code to only keep track of visible boundary edges.

2. There is another test that can be used to test for visibility of boundary edges. The first three points do not only determine the first triangle, they also determine the orientation of that triangle. If we are carful, that orientation (essentially the order of vertices in B) will be maintained at each step. The edge $V_i V_{i+1}$ is then visible from T if and only if the triangle $T V_i V_{i+1}$ is oriented consistently with the currently constructed polygon. Use this to determine visible edges (and, hence, visible vertices).

3. Line sweep can be done in any direction not only in the direction of the x-axis. Add the *sweep direction* as an argument to the Julia `line_sweep` function. The sweep direction is given by a (not-necessarily normalized) two-component vector $\mathbf{e}$. (In case of a sweep in the direction of the x-axis this vector is $\mathbf{e} = [1, 0]^\mathsf{T}$.)