Homework 1

1 Left Turn and Convexity

(a) Derive and verify that, for the given points $A = (x_a, y_a)$, $B = (x_b, y_b)$, and $C = (x_c, y_c)$ (in this order), and determinant

$$D = \begin{vmatrix} x_a & y_a & 1 \\ x_b & y_b & 1 \\ x_c & y_c & 1 \end{vmatrix},$$

A, B, and C appear in counterclockwise (ccw) order on the boundary of ΔABC if and only if D is positive (i.e. A, B, C form a left turn if $D > 0$).

The determinant can be calculated by:

$$\begin{aligned} D &= x_a \begin{vmatrix} y_b & 1 \\ y_c & 1 \end{vmatrix} - y_a \begin{vmatrix} x_b & 1 \\ x_c & 1 \end{vmatrix} + 1 \begin{vmatrix} x_b & y_b \\ x_c & y_c \end{vmatrix} && (1) \\ &= x_a(y_b - y_c) - y_a(x_b - x_c) + (x_b y_c - y_b x_c) && (2) \\ &= (x_a y_b - x_a y_c) + (x_c y_a - x_b y_a) + (x_b y_c - y_b x_c) && (3) \end{aligned}$$

Notice that the result on line (3) is the result of the cross product of the two vectors $\overrightarrow{AB}$ and $\overrightarrow{BC}$, i.e. $\overrightarrow{AB} \times \overrightarrow{BC}$. I show the determinant method of finding the cross product:

$$\overrightarrow{AB} = \langle x_b - x_a, y_b - y_a, 0 \rangle$$
$$\overrightarrow{BC} = \langle x_c - x_b, y_c - y_b, 0 \rangle$$

$$\begin{aligned} \overrightarrow{AB} \times \overrightarrow{BC} &= \begin{vmatrix} i & j & k \\ x_b - x_a & y_b - y_a & 0 \\ x_c - x_b & y_c - y_b & 0 \end{vmatrix} \\ &= \big[(x_b - x_a)(y_c - y_b) - (y_b - y_a)(x_c - x_b)\big]k \\ &= \big[(x_a y_b - x_a y_c) + (x_c y_a - x_b y_a) + (x_b y_c - y_b x_c)\big]k \end{aligned}$$

By the right-hand rule convention, when the cross product is positive, the direction along the path from $A \to B \to C \to A$ of the boundary of ΔABC is ccw, where our C is to the "left" of directed line $\overrightarrow{AB}$; when the cross product is negative, the direction along the path from $A \to B \to C \to A$ of the boundary of ΔABC is clockwise.

Reference: [1] [2]

(b) Describe an algorithm to test if polygon P (given by a circularly linked list of its n vertices in order) is convex.

From class, a convex polygon is such that, walking along the boundary of the polygon, all angles "turn" in the same "direction", resulting in all internal angles being less than 180° and only consecutive segments intersecting.

*Let there exist a method that outputs the sign of the determinant of three given coordinate points, and call it DeterminantSign. Assume it runs $O(1)$ time by plugging in the coordinate values in the formula(s) like those mentioned in part (a).

Input: A circularly linked list of the n ordered vertices of polygon P. We will call these vertices in their sequence $p_1, p_2, ..., p_n$, where p_1 is arbitrarily selected as the "first" vertex we examine in the algorithm below, p_2 is the "next" node/vertex of p_1, etc.

Output: **true** or **false** if P is convex.

```
1. A ← p1; B ← p2; C ← p3
2. direction ← DeterminantSign(A, B, C)
3. for i ← 2 to n
4.         A ← B; B ← C; C ← C.next
5.         turn ← DeterminantSign(A, B, C)
6.         if turn ≠ direction then return false
7. return true
```

Correctness : The algorithm examines each adjacent angle of the polygon formed by three adjacent points. The direction (in terms of left/right or positive/negative) of the first arbitrary (interior) angle chosen is given by direction (line 2) and formed from 3 vertices. I show by induction that this algorithm works for all polygons of $k \leq n$ vertices, where $3 \leq k \leq n$.

For $k = 3$ vertices (triangle), it is trivially true that P is a convex polygon, and the direction of the k angles match and all are acute. For $k = \{3, ..., i\}$, let all the directions of the angles up to the angle about p_{i-1} (i.e. $\angle p_{i-2}, p_{i-1}, p_i$) match the first, thus far, and therefore P is convex. The next angle formed with the $i \rightarrow \mathsf{next}^{th}$ vertex, given by $\angle p_{i-1}, p_i, p_{i\rightarrow\mathsf{next}}$, will have a turn that either does or doesn't match direction. If it matches, P is convex because all turns are the same. If it doesn't, P is not convex because the angle makes it either a concave polygon or complex polygon.

Complexity : Let accessing the next adjacent node in the linked list be a $O(1)$ time operation. Each line in the algorithm runs $O(1)$ time except for the for loop in lines 3 - 6. The for loop examines the "turn" of only each adjacent triple on the list of vertices. Therefore, the for loop runs $O(n)$ time. We are not concerned with any sorting of the vertices here. Therefore, the total runtime of the algorithm is $O(n)$.

2 Two

Two polygons share a line of support if each polygon is entirely to one side of the line (tangent), and at least one point from each polygon is on that line.

I show that the number of these lines is at most $2 \times \mathtt{min}(n_1, n_2)$.

There is at least two lines of support shared by the two polygon that would form the edges of the convex hull of the two disjointed polygons.

3 Three

References

[1] Cross Product of Two Vectors: https://www.cuemath.com/geometry/cross-product/

[2] Math World - Curve Orientation: https://mathworld.wolfram.com/CurveOrientation.html