Computational Geometry

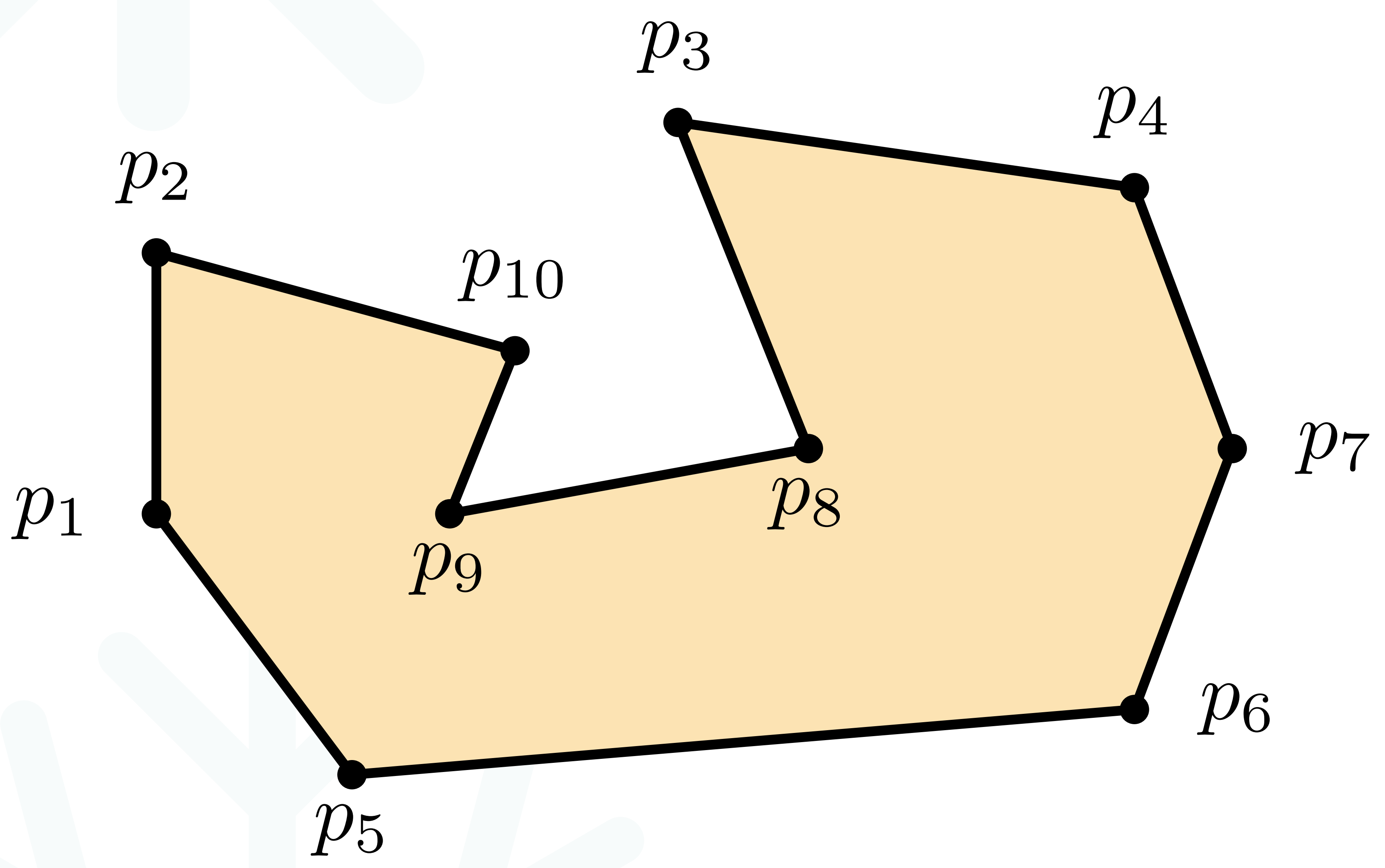
Tutorial #3 — Jordan Curves and Polygons

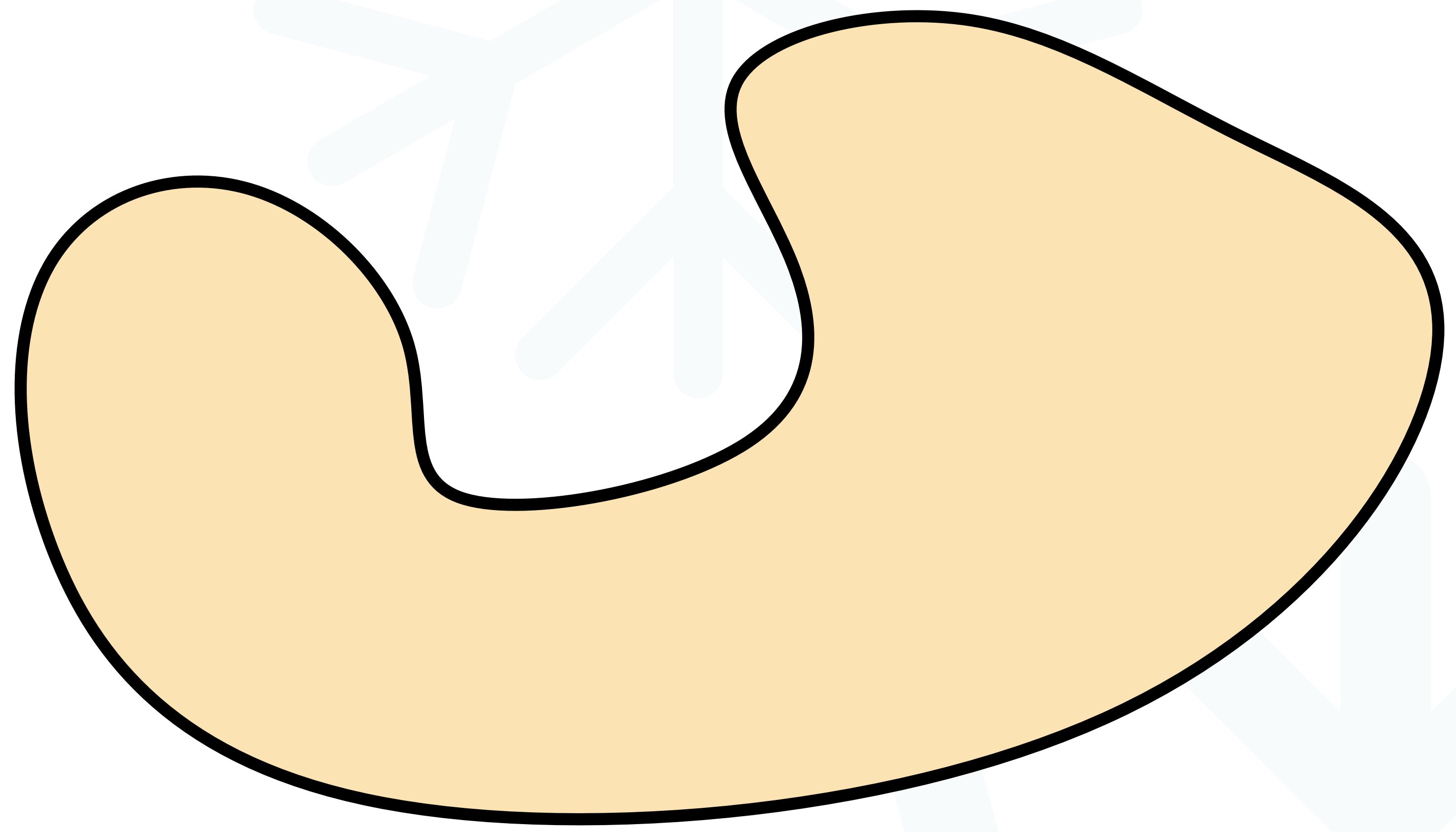
Institut für Betriebssysteme
und Rechnerverbund
Algorithmik

Simple (Plane) Curves

Generalising Polygons

Curves in the plane

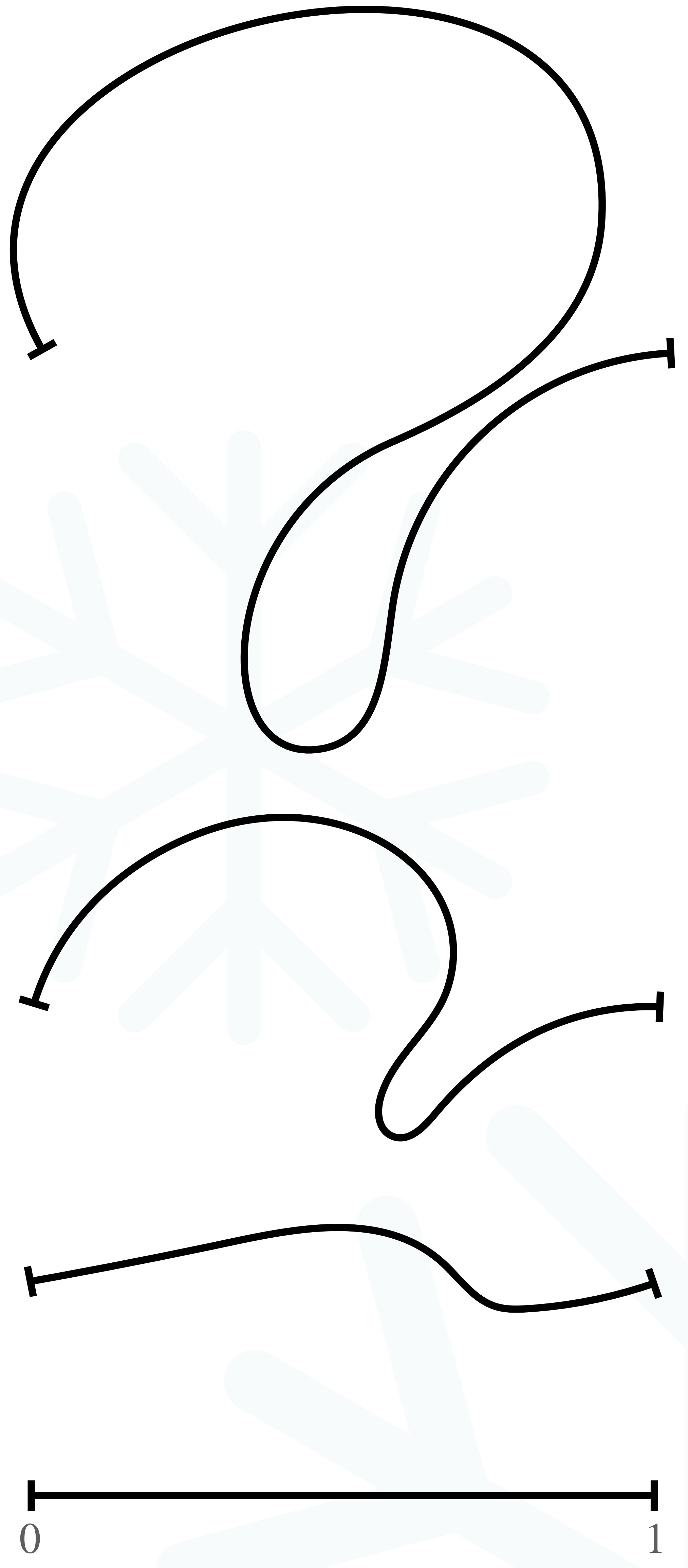




Simple paths in the plane

- Two sets are *homeomorphic* if we can define a continuous, bijective mapping $h : X \to Y$ between the two.
- A *simple path* is a subset of the plane that is homeomorphic to the interval $[0,1] \subset \mathbb{R}$, i.e., $h : [0,1] \to L \subset \mathbb{R}^2$.

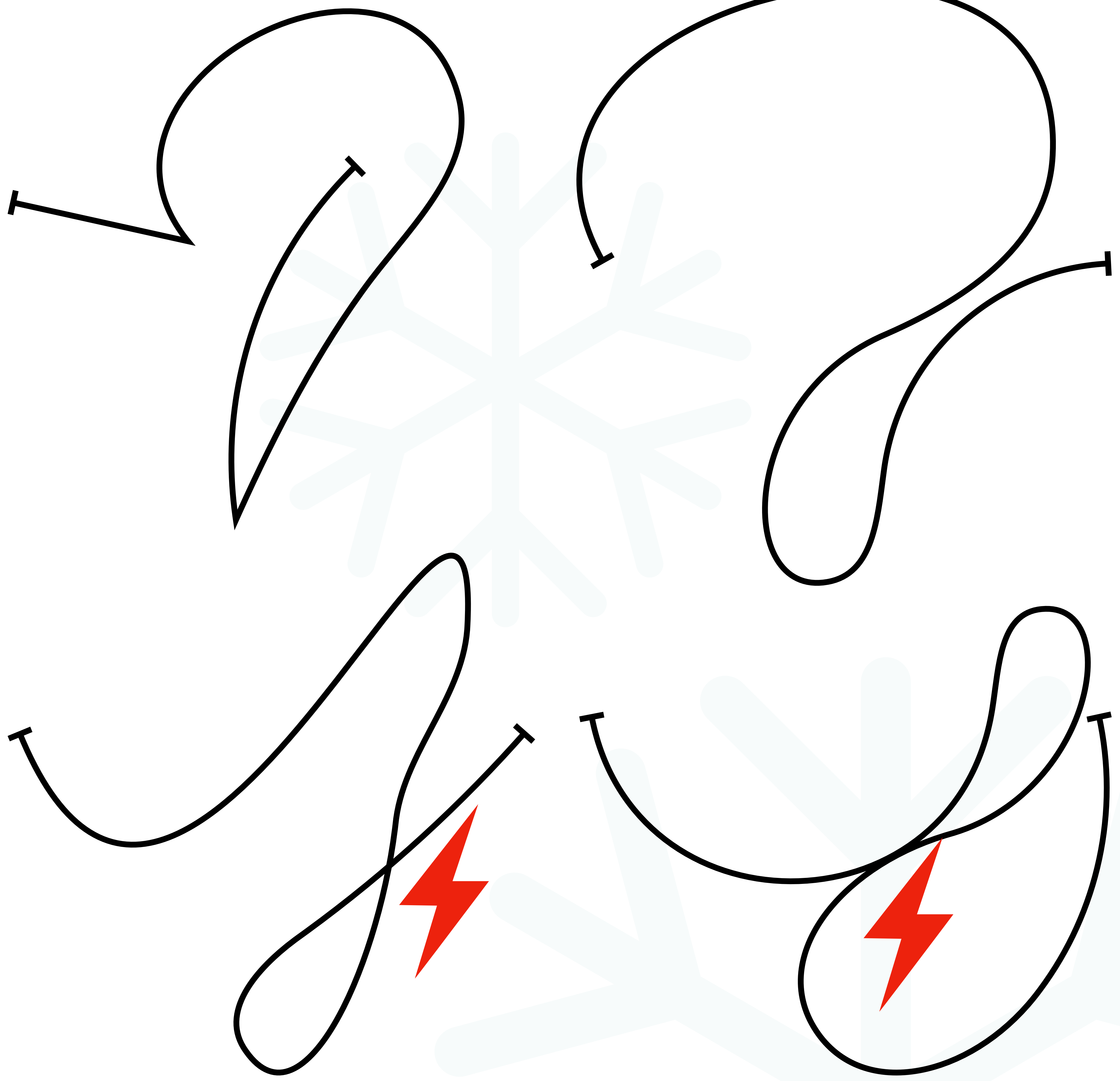
Geometrically, imagine any shape that can be continuously “morphed” from a straight line.



Simple paths in the plane

- Two sets are *homeomorphic* if we can define a continuous, bijective mapping $h : X \to Y$ between the two.
- A *simple path* is a subset of the plane that is homeomorphic to the interval $[0,1] \subset \mathbb{R}$, i.e., $h : [0,1] \to L \subset \mathbb{R}^2$.

Geometrically, imagine any shape that can be continuously "morphed" from a straight line.



Simple paths in the plane

Example: Straight line segments

- The line segment between two points $p, q \in \mathbb{R}^2$ is a simple curve

$$\overline{pq} = \{\, x \in \mathbb{R}^2 \quad | \quad \exists\, s \in [0,1] :\ x = p \cdot s + q \cdot (1 - s) \,\}.$$

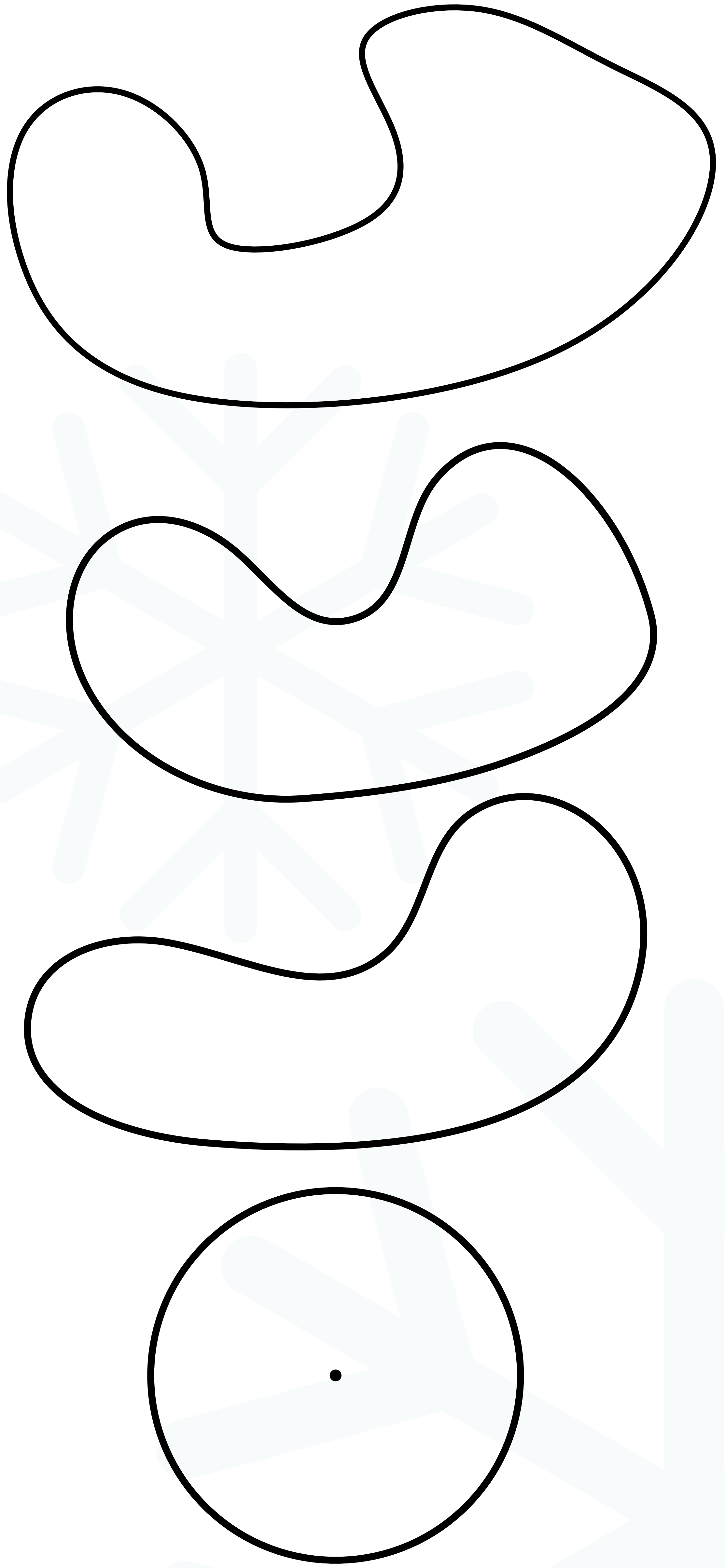
- We can define a homeomorphism between $[0,1] \subset \mathbb{R}$ and $\overline{pq}$:

$$h_{\overline{pq}} : [0,1] \to \overline{pq}, \quad s \mapsto p \cdot s + p \cdot (1 - s)$$

Closed curves in the plane

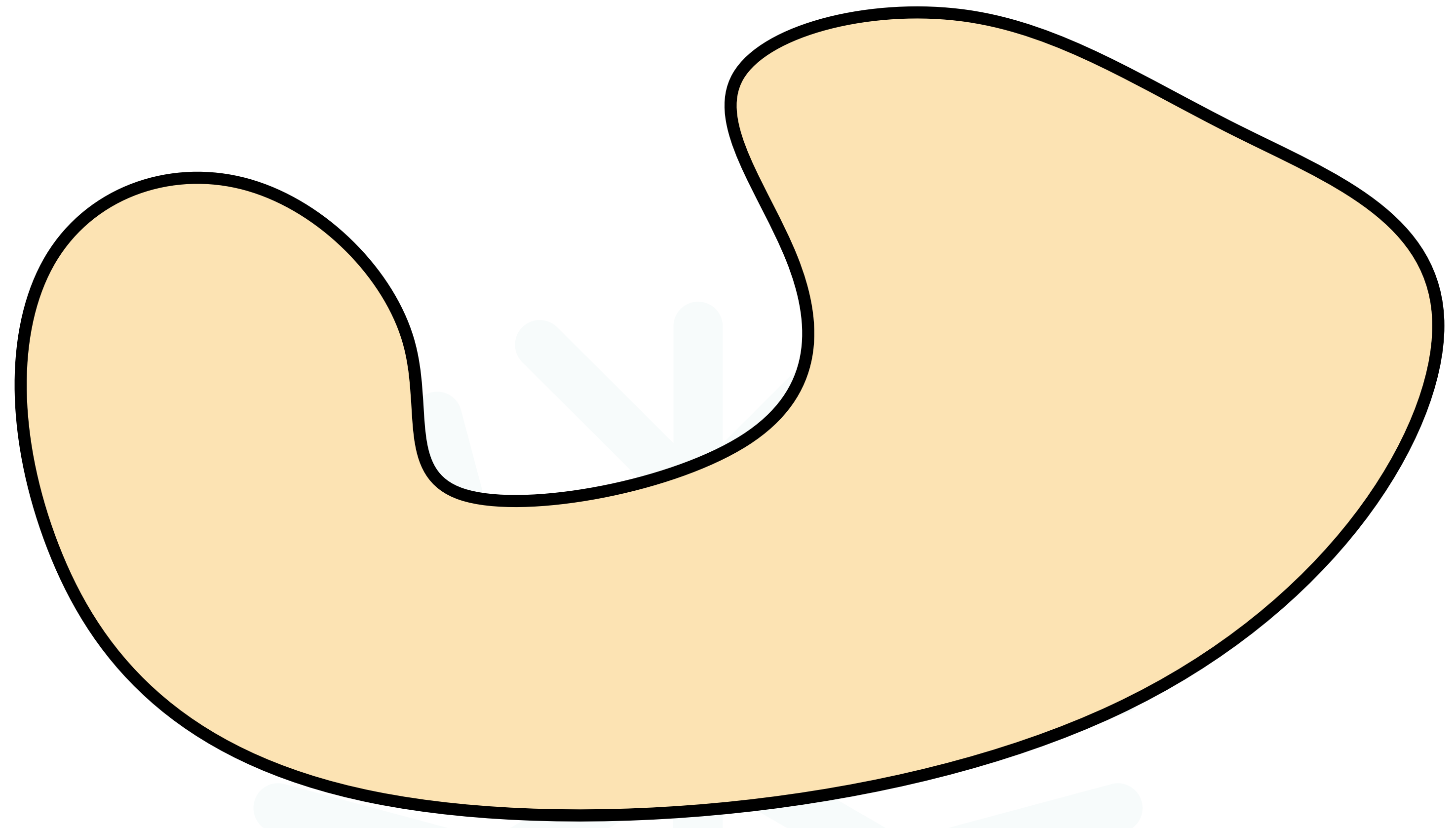
- Consider the (Euclidean) unit circle $S_1 = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}$.
- A *simple closed curve* is a point set in the plane that is homeomorphic to S_1.
- Then: $h : S_1 \to C \subset \mathbb{R}^2$.

Geometrically, imagine any shape that can be continuously “morphed” from the circle.

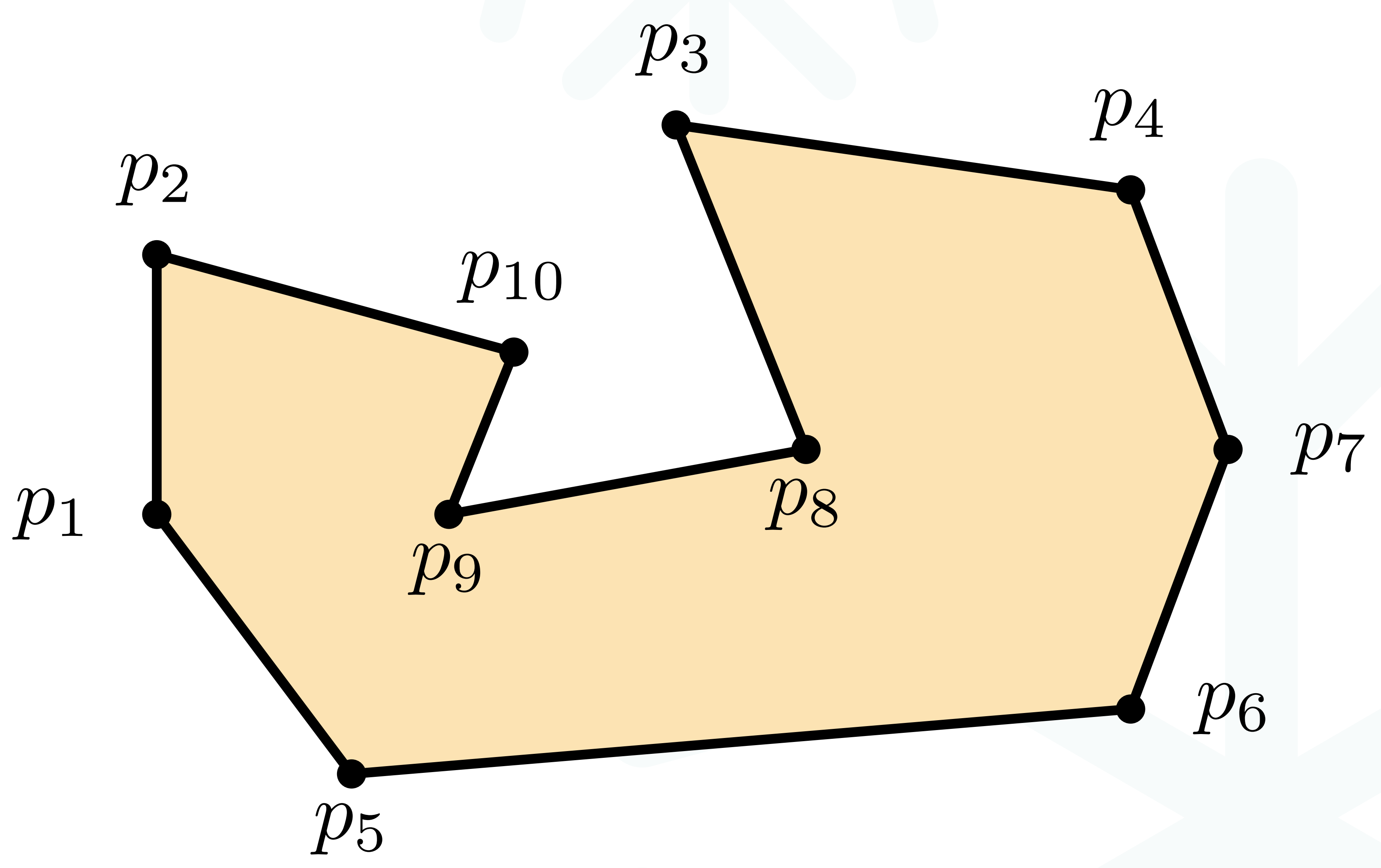


Closed curves in the plane

- Consider the (Euclidean) unit circle $S_1 = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}$.
- A *simple closed curve* is a point set in the plane that is homeomorphic to S_1.



- A *simple* polygon P is a *closed curve* composed of a finite number of line segments.



Closed curves in the plane

Example: Simple polygons

- Recall that line segments are simple paths.
- If two simple paths meet only in a common endpoint, we can join them into a longer path.
- By induction, we can join the line segments that define a simple polygon P into a closed curve by joining them in CCW order.

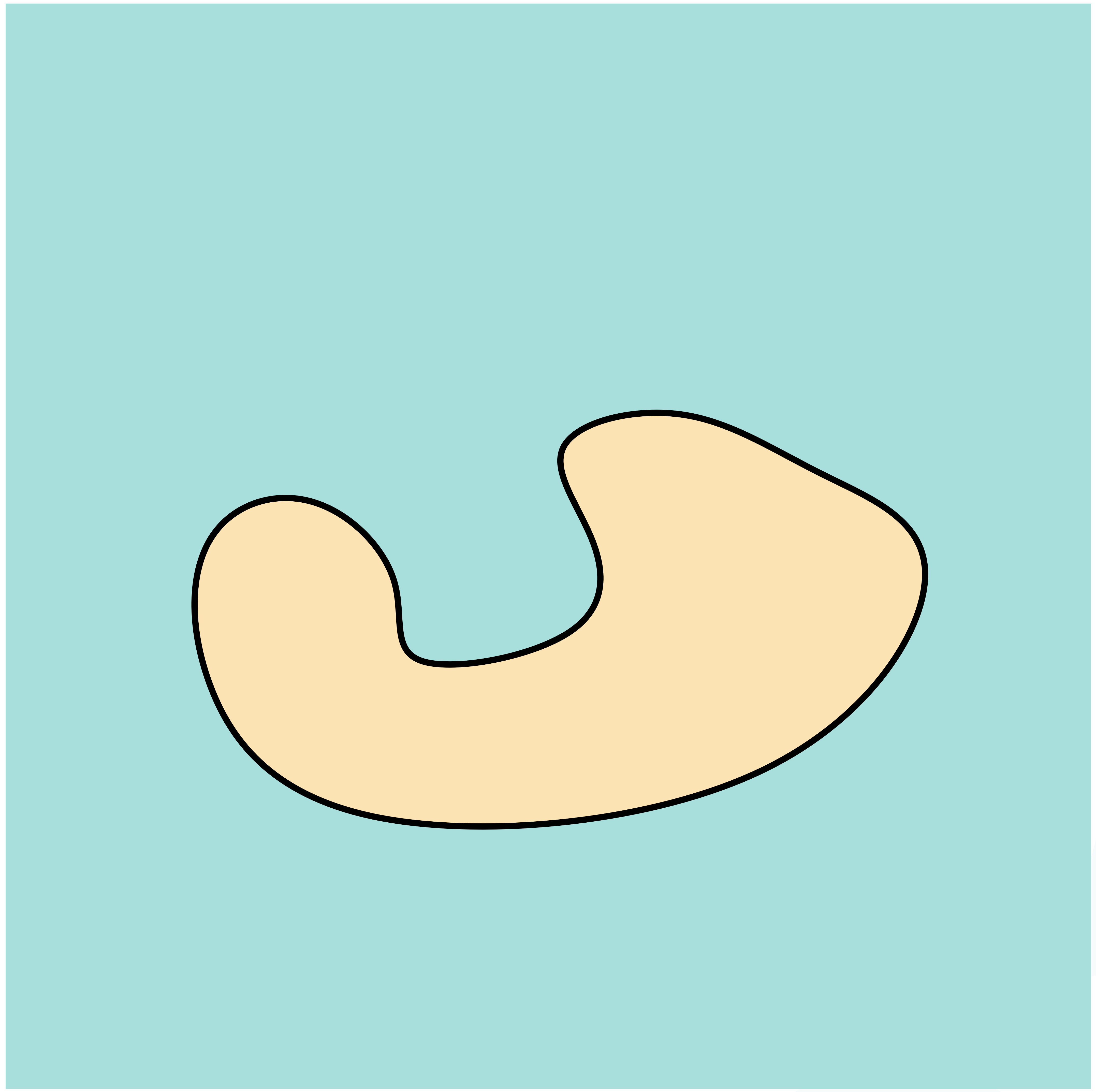
Jordan Curves

Author and Date

Jordan Curve Theorem

Theorem E3.1 (Jordan Curves)

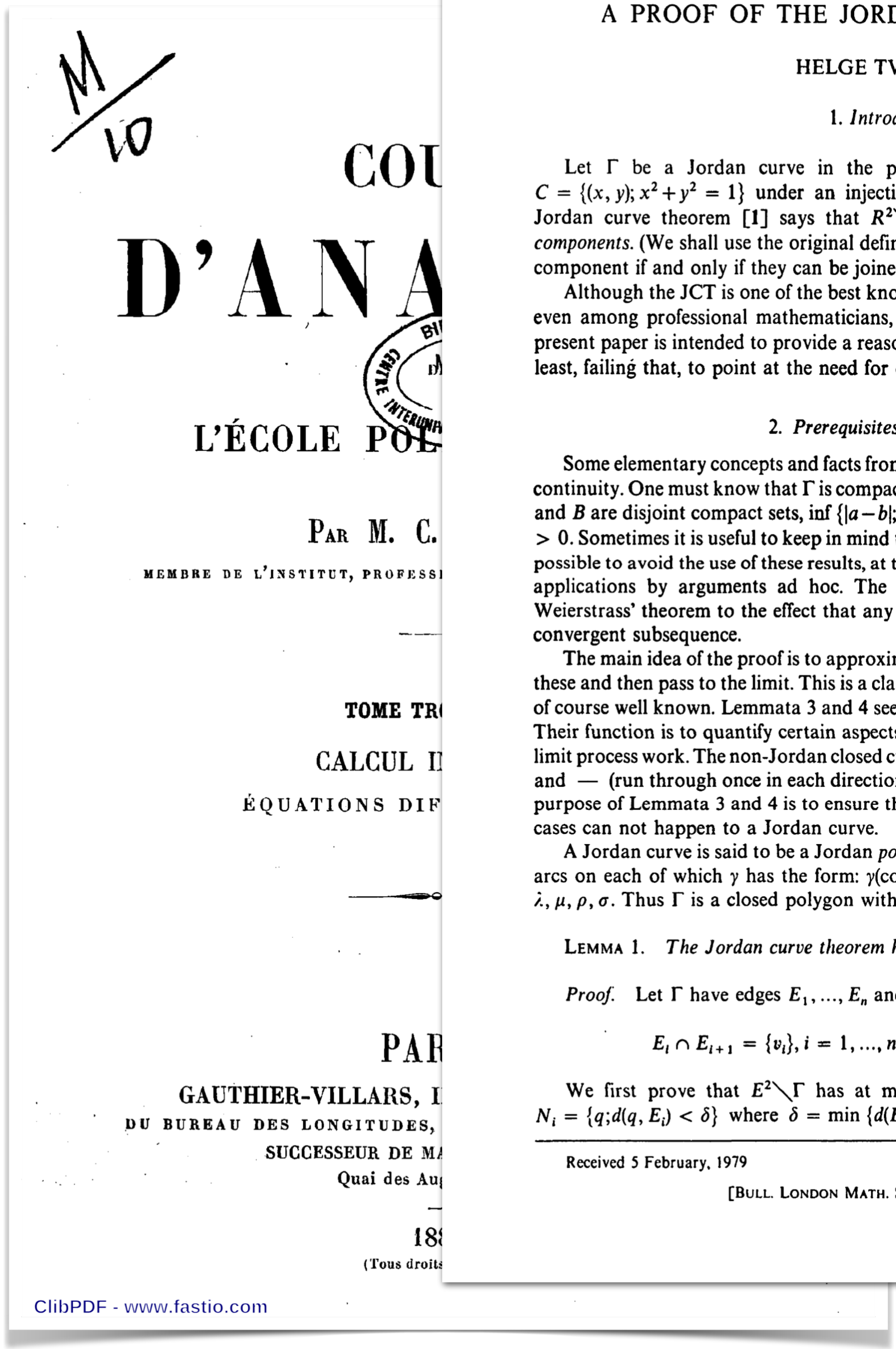
Let C be a Jordan curve in the plane $\mathbb{R}^2$. Then its complement $\mathbb{R}^2 \setminus C$ consists of exactly two connected components: the bounded ***interior*** *and the unbounded* ***exterior***.



Jordan Curve Theorem

Theorem E3.1 (Jordan Curves)

Let C be a Jordan curve in the plane $\mathbb{R}^2$. Then its complement $\mathbb{R}^2 \setminus C$ consists of exactly two connected components: the bounded ***interior*** *and the unbounded* ***exterior****.*



M/10

COU

D'ANA

L'ÉCOLE POL

PAR M. C.

MEMBRE DE L'INSTITUT, PROFESS

TOME TR

CALCUL I

ÉQUATIONS DIF

PAR

GAUTHIER-VILLARS, I

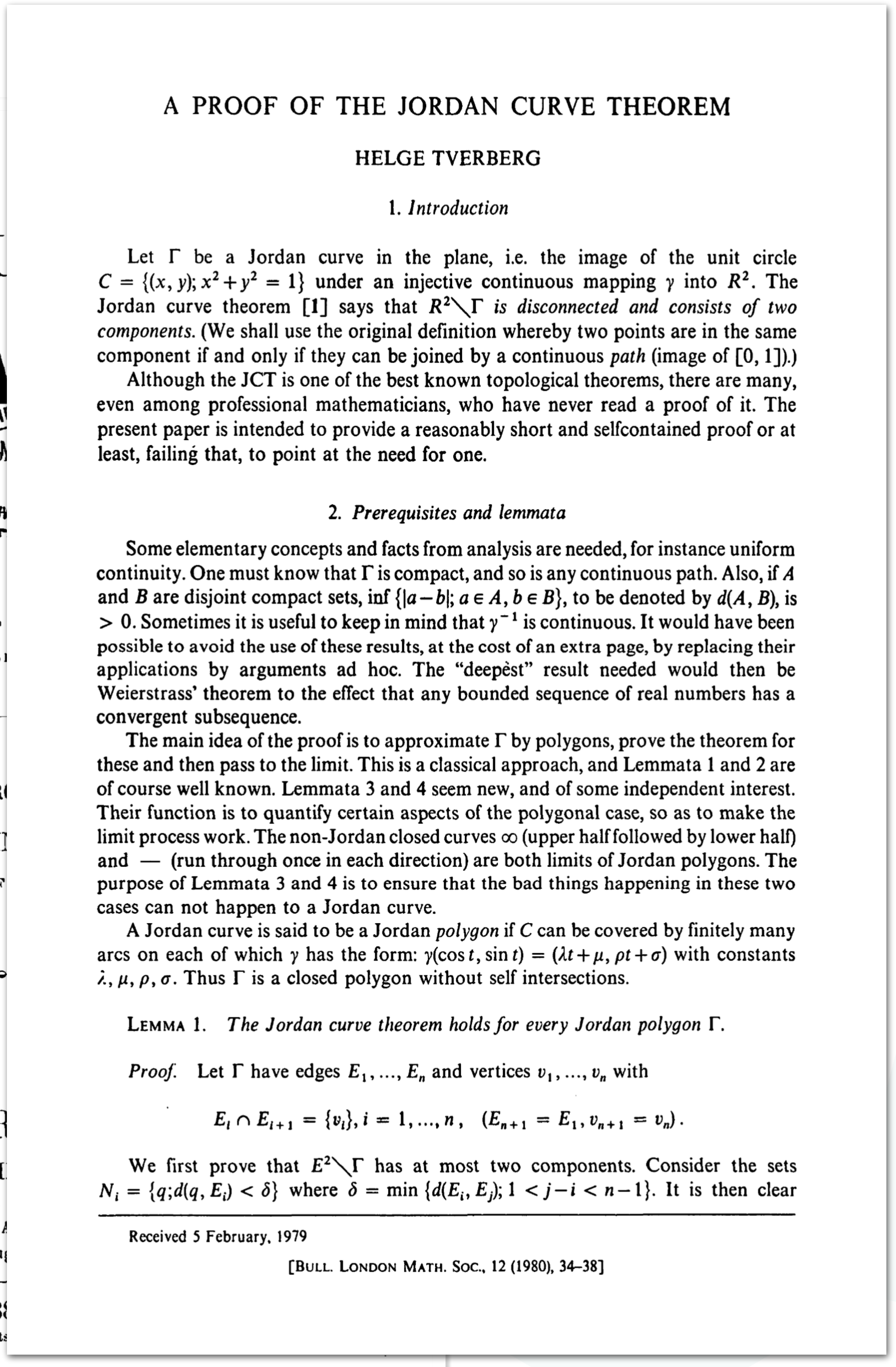
DU BUREAU DES LONGITUDES,

SUCCESSEUR DE MA

Quai des Au

18

(Tous droits



A PROOF OF THE JORDAN CURVE THEOREM

HELGE TVERBERG

1. *Introduction*

Let Γ be a Jordan curve in the plane, i.e. the image of the unit circle $C = \{(x, y); x^2 + y^2 = 1\}$ under an injective continuous mapping γ into R^2. The Jordan curve theorem [1] says that $R^2 \setminus \Gamma$ *is disconnected and consists of two components*. (We shall use the original definition whereby two points are in the same component if and only if they can be joined by a continuous *path* (image of $[0, 1]$).)

Although the JCT is one of the best known topological theorems, there are many, even among professional mathematicians, who have never read a proof of it. The present paper is intended to provide a reasonably short and selfcontained proof or at least, failing that, to point at the need for one.

2. *Prerequisites and lemmata*

Some elementary concepts and facts from analysis are needed, for instance uniform continuity. One must know that Γ is compact, and so is any continuous path. Also, if A and B are disjoint compact sets, $\inf \{|a-b|; a \in A, b \in B\}$, to be denoted by $d(A, B)$, is > 0. Sometimes it is useful to keep in mind that γ^{-1} is continuous. It would have been possible to avoid the use of these results, at the cost of an extra page, by replacing their applications by arguments ad hoc. The "deepèst" result needed would then be Weierstrass' theorem to the effect that any bounded sequence of real numbers has a convergent subsequence.

The main idea of the proof is to approximate Γ by polygons, prove the theorem for these and then pass to the limit. This is a classical approach, and Lemmata 1 and 2 are of course well known. Lemmata 3 and 4 seem new, and of some independent interest. Their function is to quantify certain aspects of the polygonal case, so as to make the limit process work. The non-Jordan closed curves ∞ (upper half followed by lower half) and — (run through once in each direction) are both limits of Jordan polygons. The purpose of Lemmata 3 and 4 is to ensure that the bad things happening in these two cases can not happen to a Jordan curve.

A Jordan curve is said to be a Jordan *polygon* if C can be covered by finitely many arcs on each of which γ has the form: $\gamma(\cos t, \sin t) = (\lambda t + \mu, \rho t + \sigma)$ with constants $\lambda, \mu, \rho, \sigma$. Thus Γ is a closed polygon without self intersections.

LEMMA 1. *The Jordan curve theorem holds for every Jordan polygon Γ.*

Proof. Let Γ have edges $E_1, \ldots, E_n$ and vertices $v_1, \ldots, v_n$ with

$$E_i \cap E_{i+1} = \{v_i\}, i = 1, \ldots, n, \quad (E_{n+1} = E_1, v_{n+1} = v_n).$$

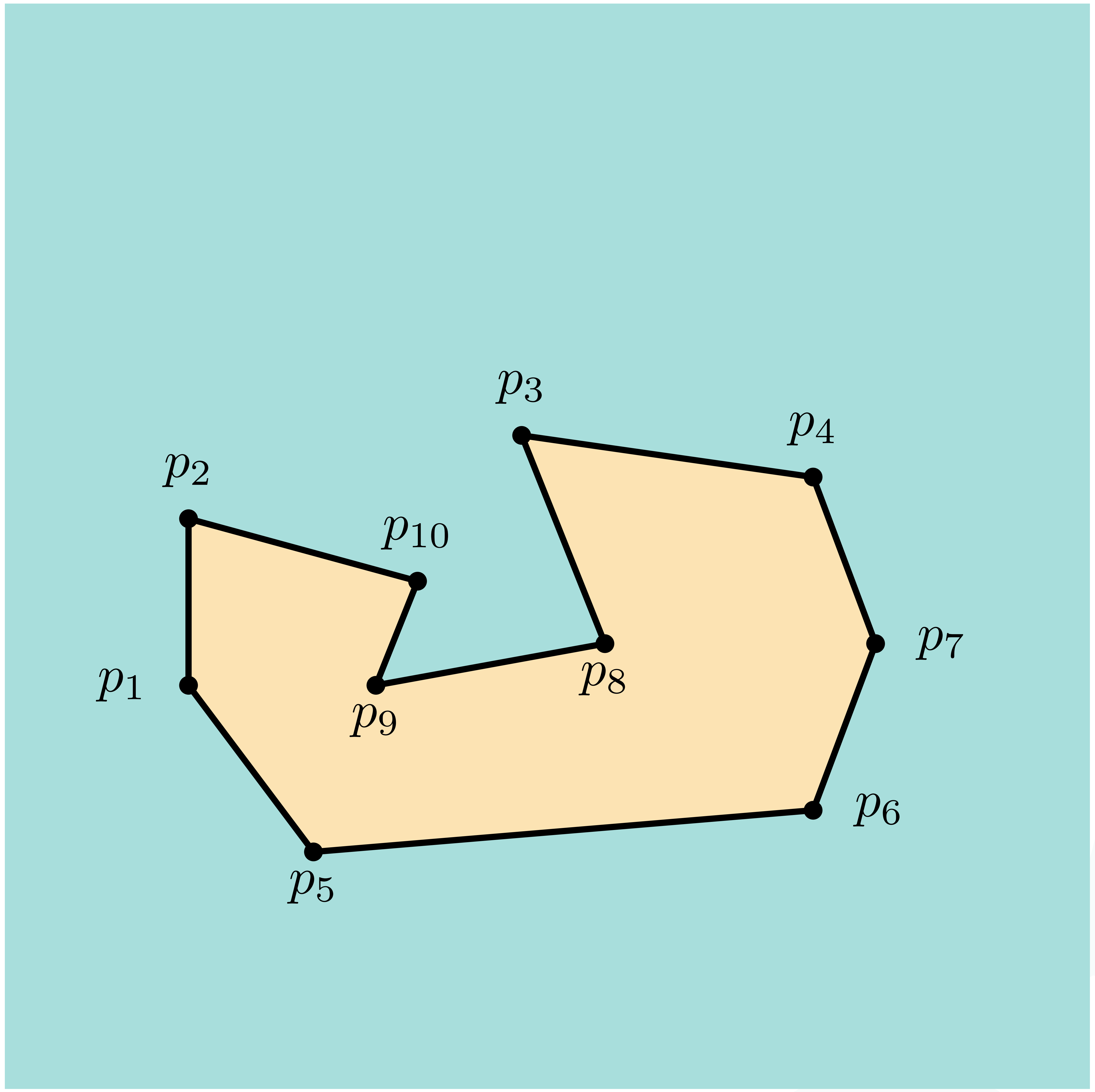
We first prove that $E^2 \setminus \Gamma$ has at most two components. Consider the sets $N_i = \{q; d(q, E_i) < \delta\}$ where $\delta = \min \{d(E_i, E_j); 1 < j-i < n-1\}$. It is then clear

Received 5 February, 1979

Jordan Curve Theorem

Theorem E3.1 (Jordan Polygons)

Let P be a simple polygon in the plane $\mathbb{R}^2$. Then its complement $\mathbb{R}^2 \setminus P$ consists of exactly two connected components: the bounded ***interior*** *and the unbounded* ***exterior***.

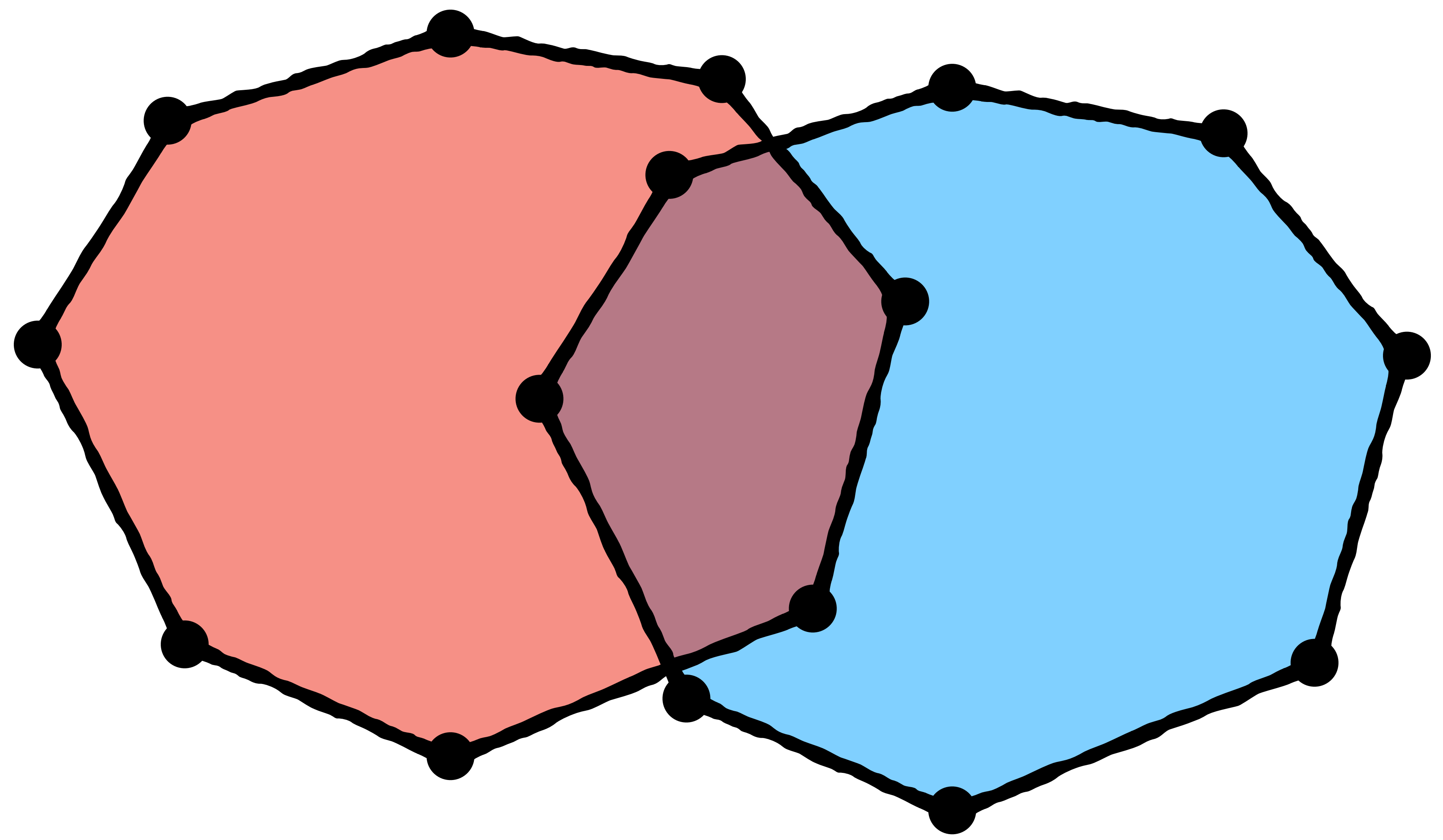


Boolean Operations

Boolean Operations on Convex Polygons

Given two convex Polygons P and Q, we seek to determine:

$$P \cap Q, P \cup Q, P \setminus Q, (Q \setminus P)$$



Naive Algorithm using these tools

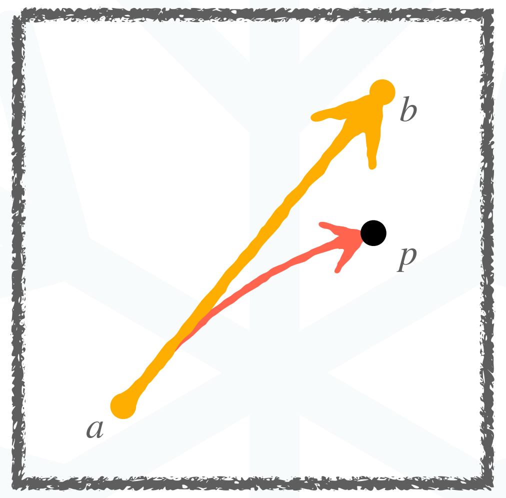
Eliminate/Add points, then recompute convex hull

Given: Convex polygons $P := p_1, \ldots, p_n$ and $Q := q_1, \ldots, q_m$.

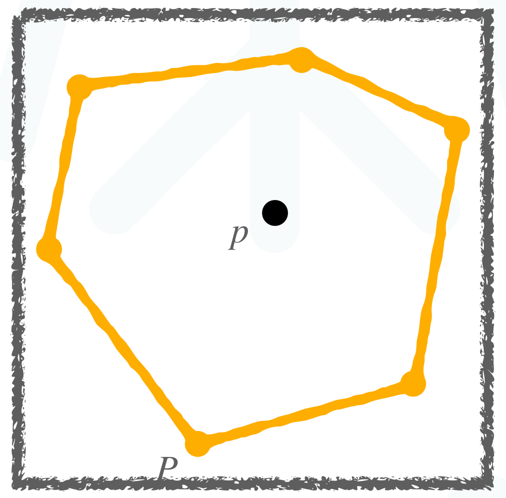
Wanted: The convex polygon $P \cap Q$.

Idea: Determine extreme points of $P \cap Q$ in $\mathcal{O}(n^2)$, then compute the convex hull in $\mathcal{O}(n \log h)$.

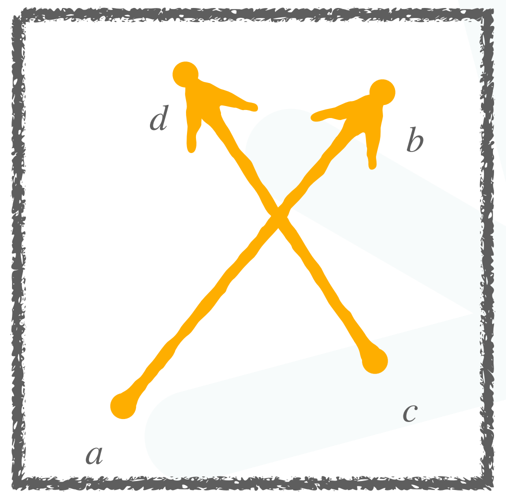
This gives us an $\mathcal{O}(n^2)$-algorithm.



$\mathcal{O}(1)$



$\mathcal{O}(\log n)$



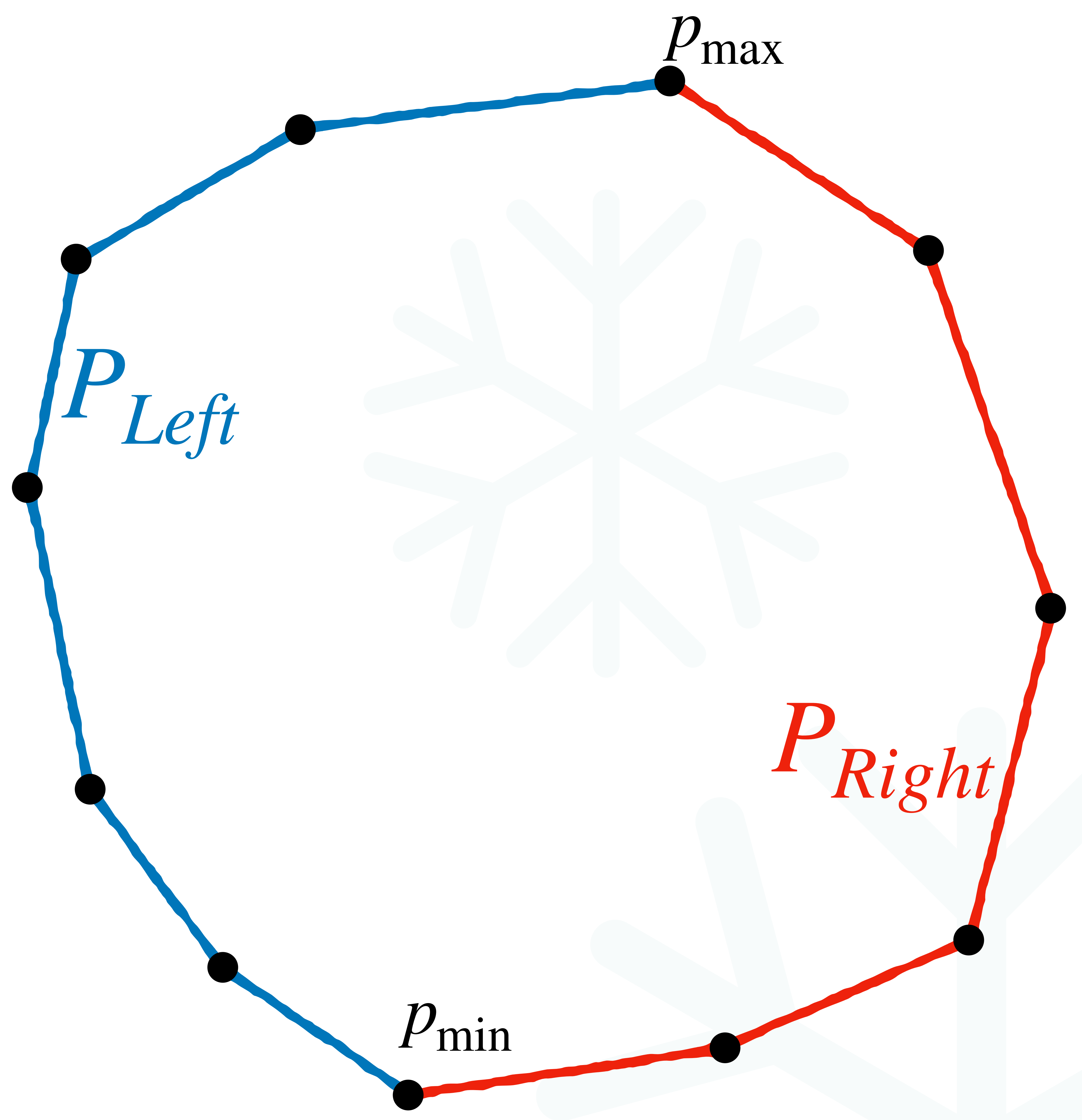
$\mathcal{O}(1)$

Convex polygons

… can be decomposed!

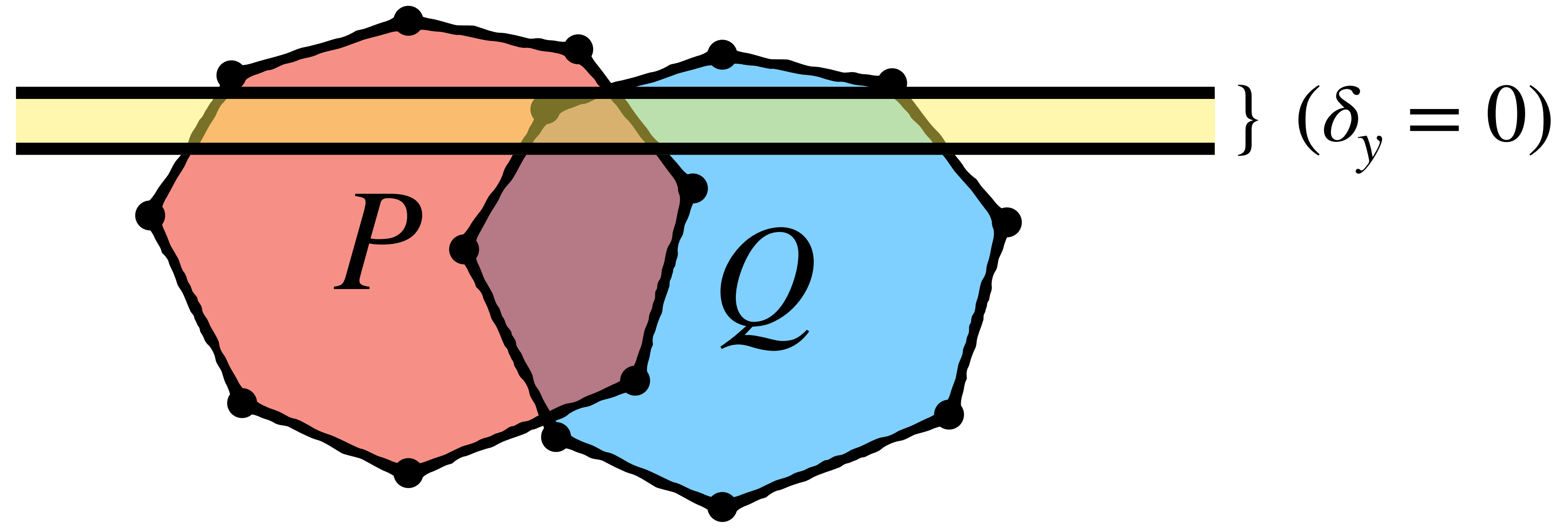
- Given $P = p_1, \ldots, p_n$
- Compute $p_{\min}$ and $p_{\max}$ in $\mathcal{O}(n)$ along the y-axis
- We obtain P_L and P_R:

$$P_L = \quad p_{\max}, \ldots, p_{\min}$$
$$P_R = \quad p_{\min}, \ldots, p_{\max}$$

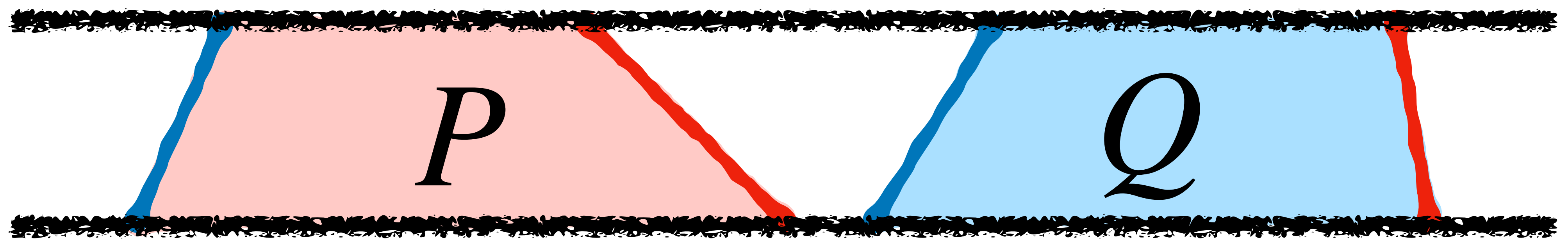


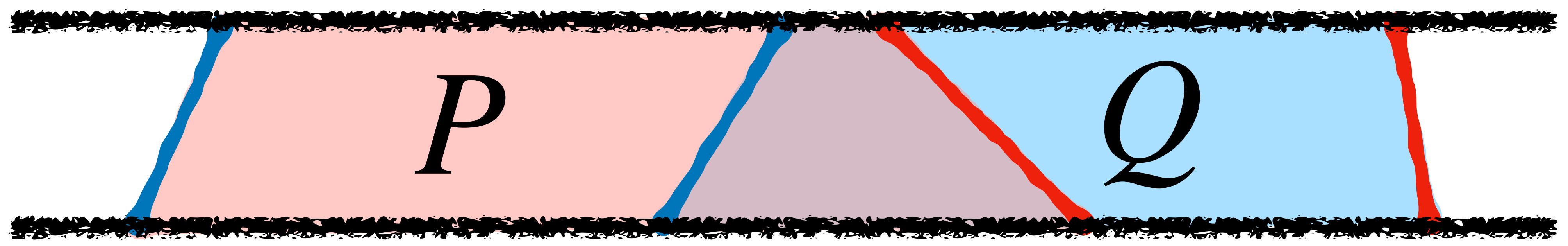
Towards an $\mathcal{O}(n)$-Algorithm

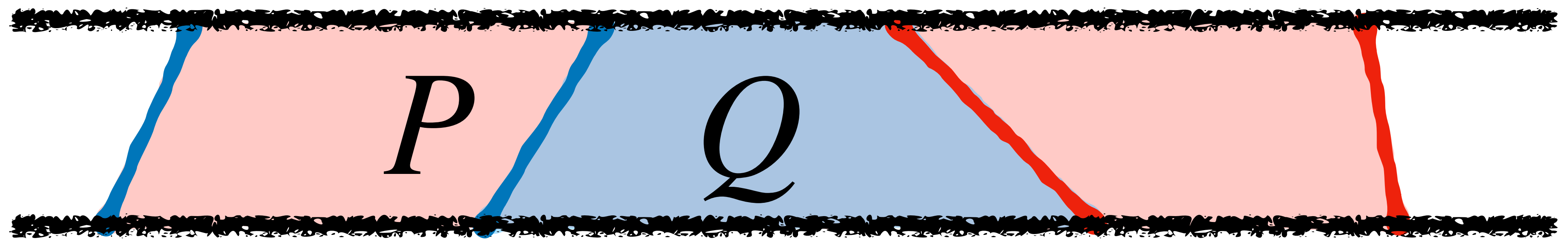
Using left- and right decomposition



- If we slice through P and Q horizontally, at some y (see right, exaggerated):
  - (a) P and Q do not intersect,
  - (b) P and Q overlap partially, or
  - (c) one contains the other.

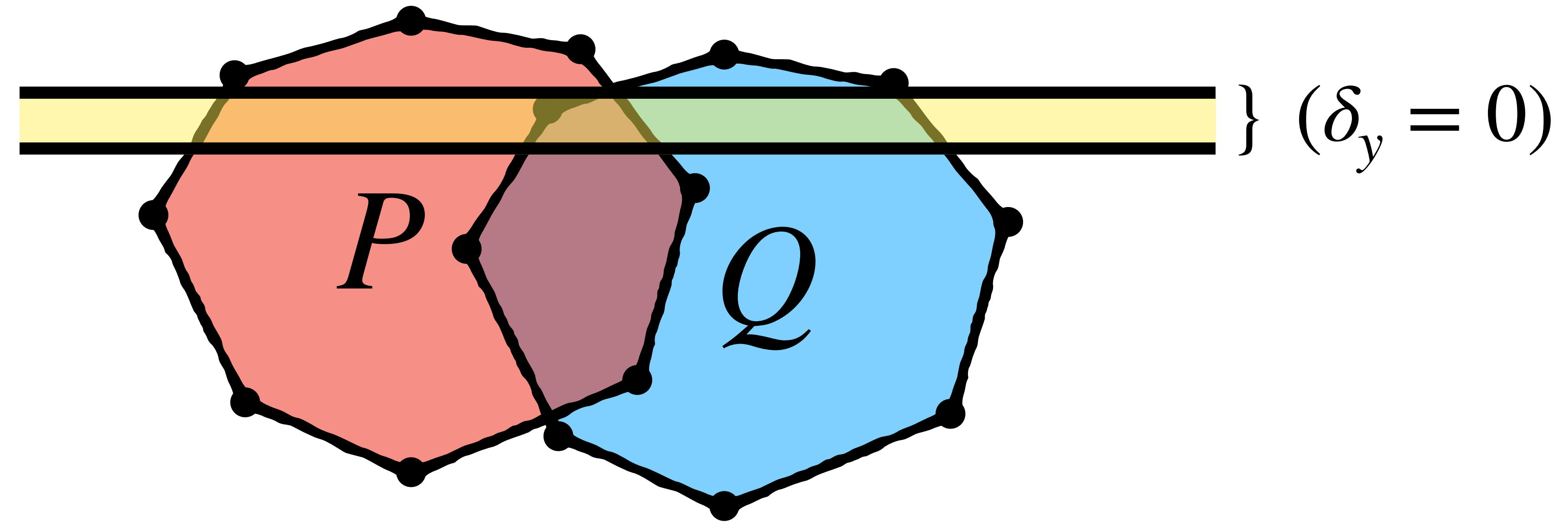




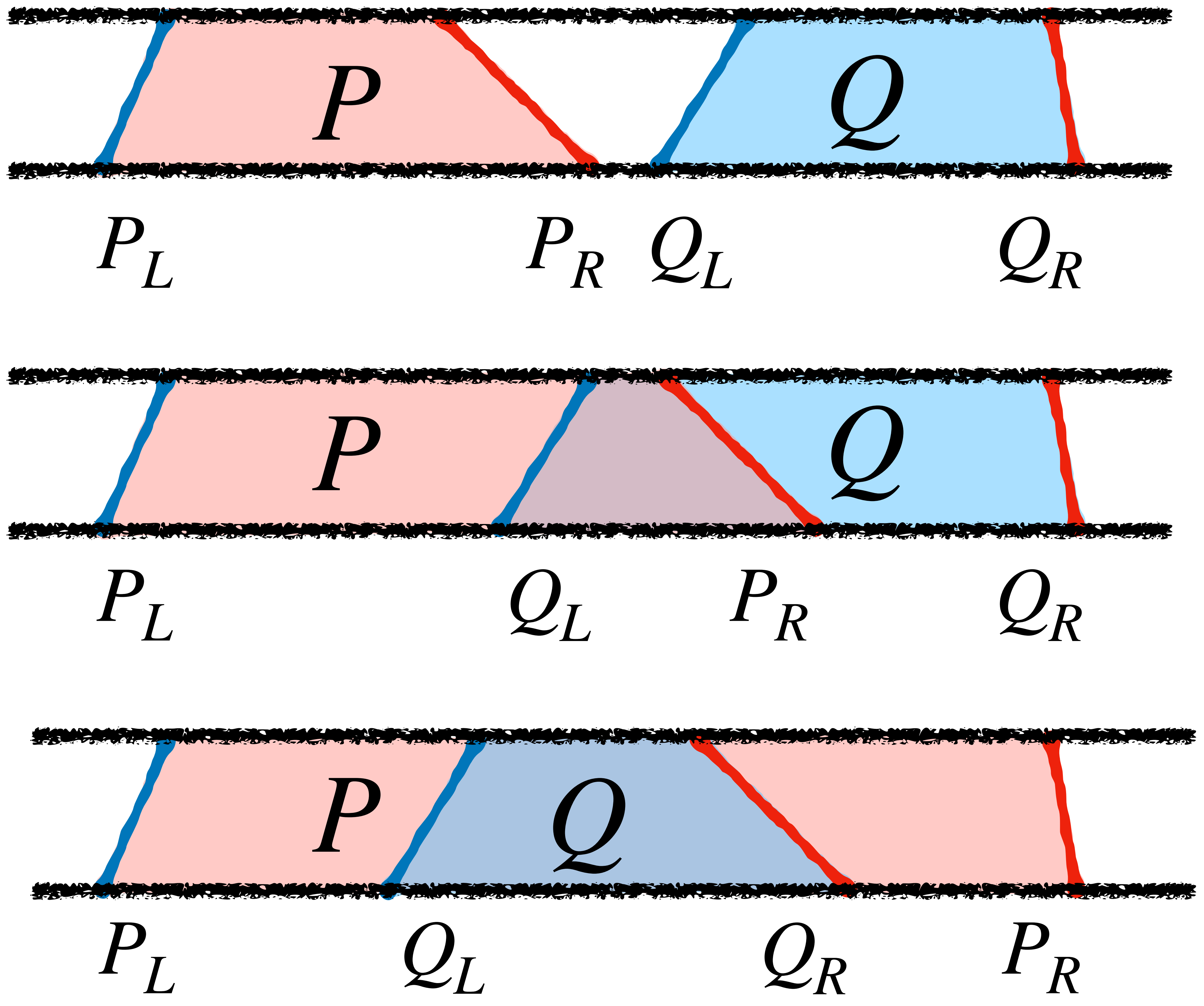


Towards an $\mathcal{O}(n)$-Algorithm

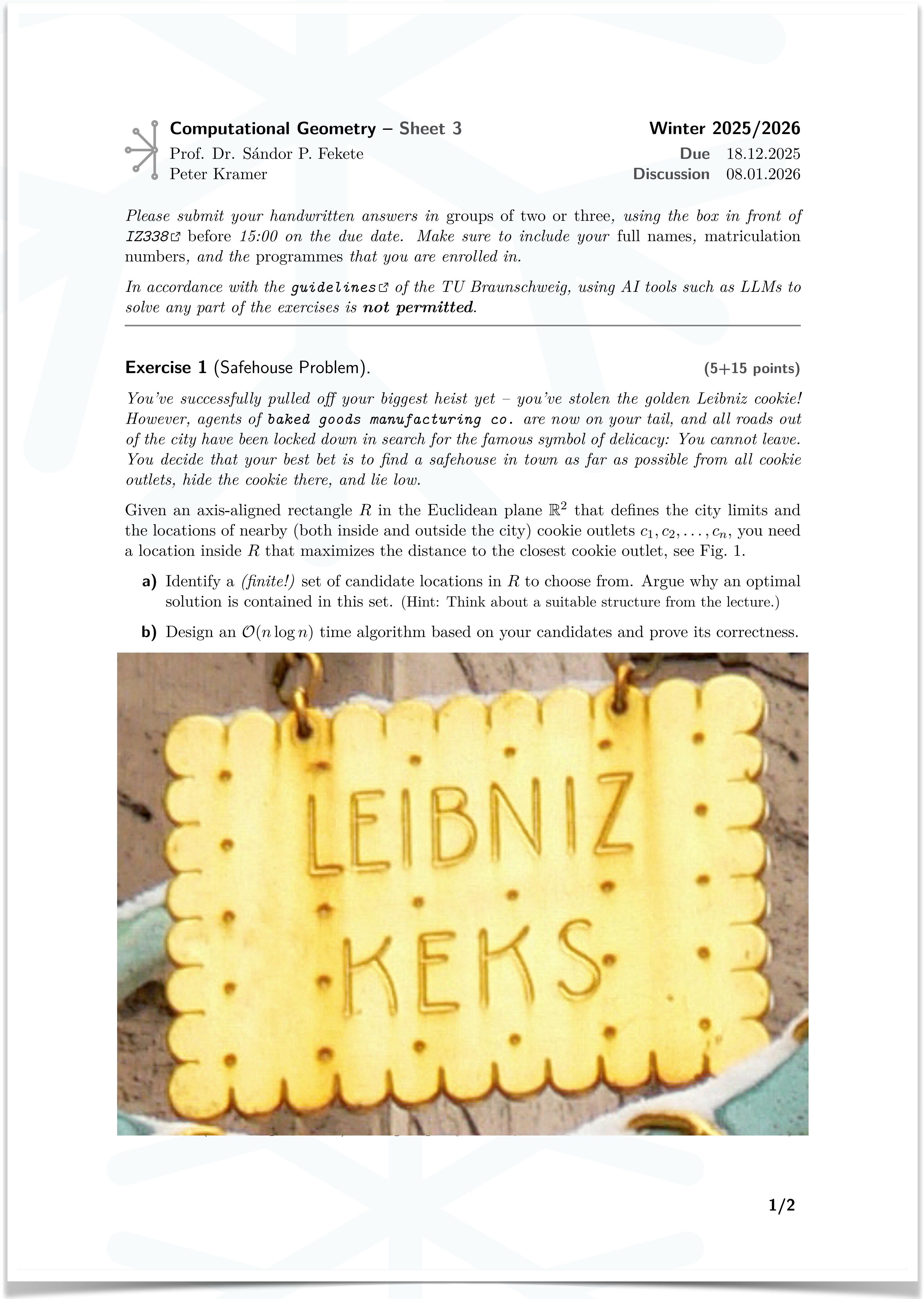
Using left- and right decomposition



- If we slice through P and Q horizontally, at some y (see right, exaggerated):
  - (a) P and Q do not intersect,
  - (b) P and Q overlap partially, or
  - (c) one contains the other.
- Each case corresponds to an x-order of the chains at that y-coordinate.



Homework Sheet #3



Computational Geometry – Sheet 3 **Winter 2025/2026**

Prof. Dr. Sándor P. Fekete **Due** 18.12.2025

Peter Kramer **Discussion** 08.01.2026

Please submit your handwritten answers in groups of two or three, *using the box in front of* IZ338 before 15:00 *on the due date. Make sure to include your* full names, matriculation numbers, *and the programmes that you are enrolled in.*

In accordance with the **guidelines** *of the TU Braunschweig, using AI tools such as LLMs to solve any part of the exercises is* **not permitted**.

Exercise 1 (Safehouse Problem). **(5+15 points)**

You've successfully pulled off your biggest heist yet – you've stolen the golden Leibniz cookie! However, agents of `baked goods manufacturing co.` *are now on your tail, and all roads out of the city have been locked down in search for the famous symbol of delicacy: You cannot leave. You decide that your best bet is to find a safehouse in town as far as possible from all cookie outlets, hide the cookie there, and lie low.*

Given an axis-aligned rectangle R in the Euclidean plane $\mathbb{R}^2$ that defines the city limits and the locations of nearby (both inside and outside the city) cookie outlets $c_1, c_2, \dots, c_n$, you need a location inside R that maximizes the distance to the closest cookie outlet, see Fig. 1.

a) Identify a *(finite!)* set of candidate locations in R to choose from. Argue why an optimal solution is contained in this set. (Hint: Think about a suitable structure from the lecture.)

b) Design an $\mathcal{O}(n \log n)$ time algorithm based on your candidates and prove its correctness.

1/2

Exercise 1 (Safehouse Problem). **(5+15 points)**

You've successfully pulled off your biggest heist yet – you've stolen the golden Leibniz cookie! However, agents of `baked goods manufacturing co.` *are now on your tail, and all roads out of the city have been locked down in search for the famous symbol of delicacy: You cannot leave. You decide that your best bet is to find a safehouse in town as far as possible from all cookie outlets, hide the cookie there, and lie low.*

Given an axis-aligned rectangle R in the Euclidean plane $\mathbb{R}^2$ that defines the city limits and the locations of nearby (both inside and outside the city) cookie outlets $c_1, c_2, \dots, c_n$, you need a location inside R that maximizes the distance to the closest cookie outlet, see Fig. 1.

a) Identify a *(finite!)* set of candidate locations in R to choose from. Argue why an optimal solution is contained in this set. (Hint: Think about a suitable structure from the lecture.)

b) Design an $\mathcal{O}(n \log n)$ time algorithm based on your candidates and prove its correctness.

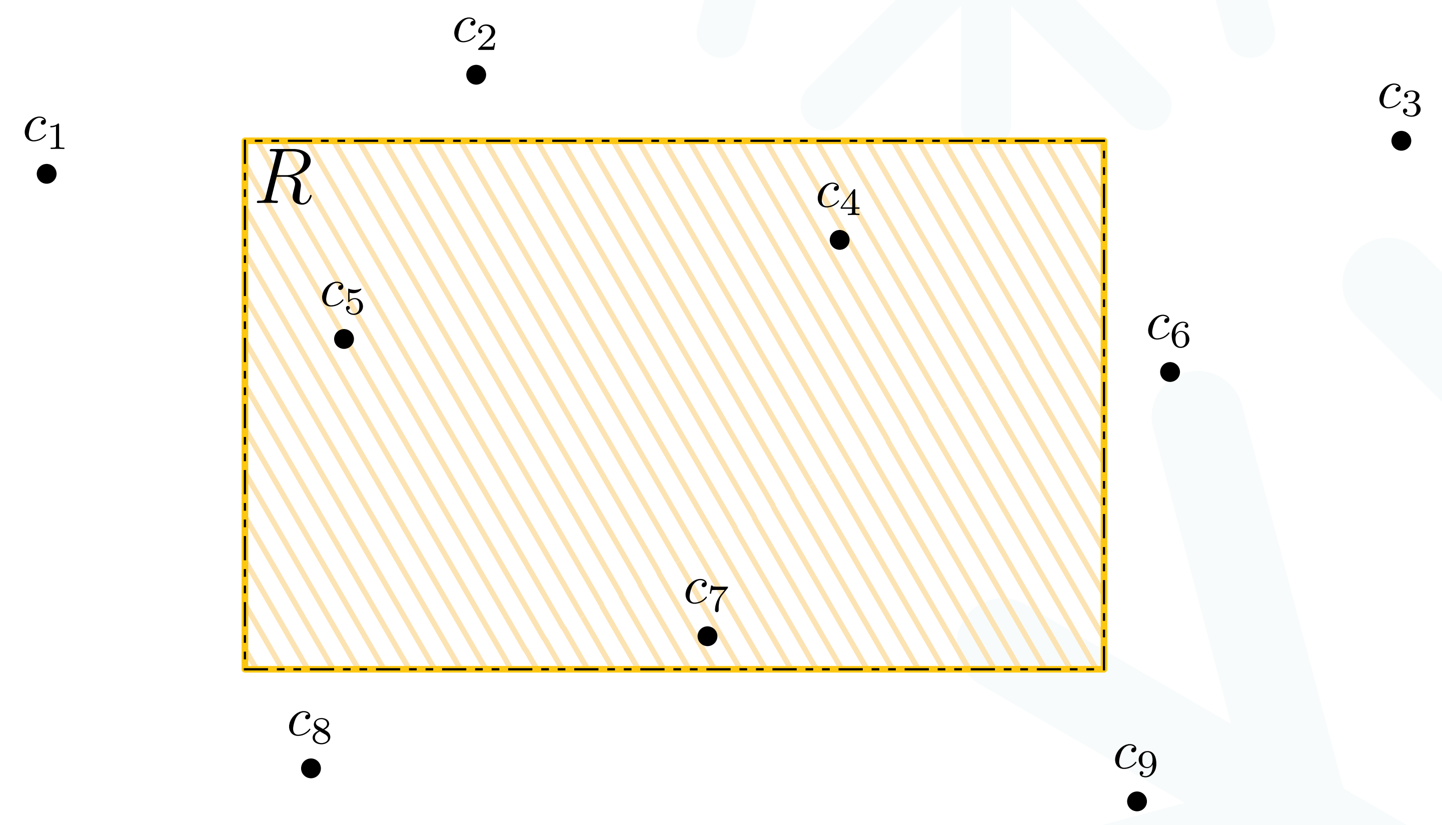
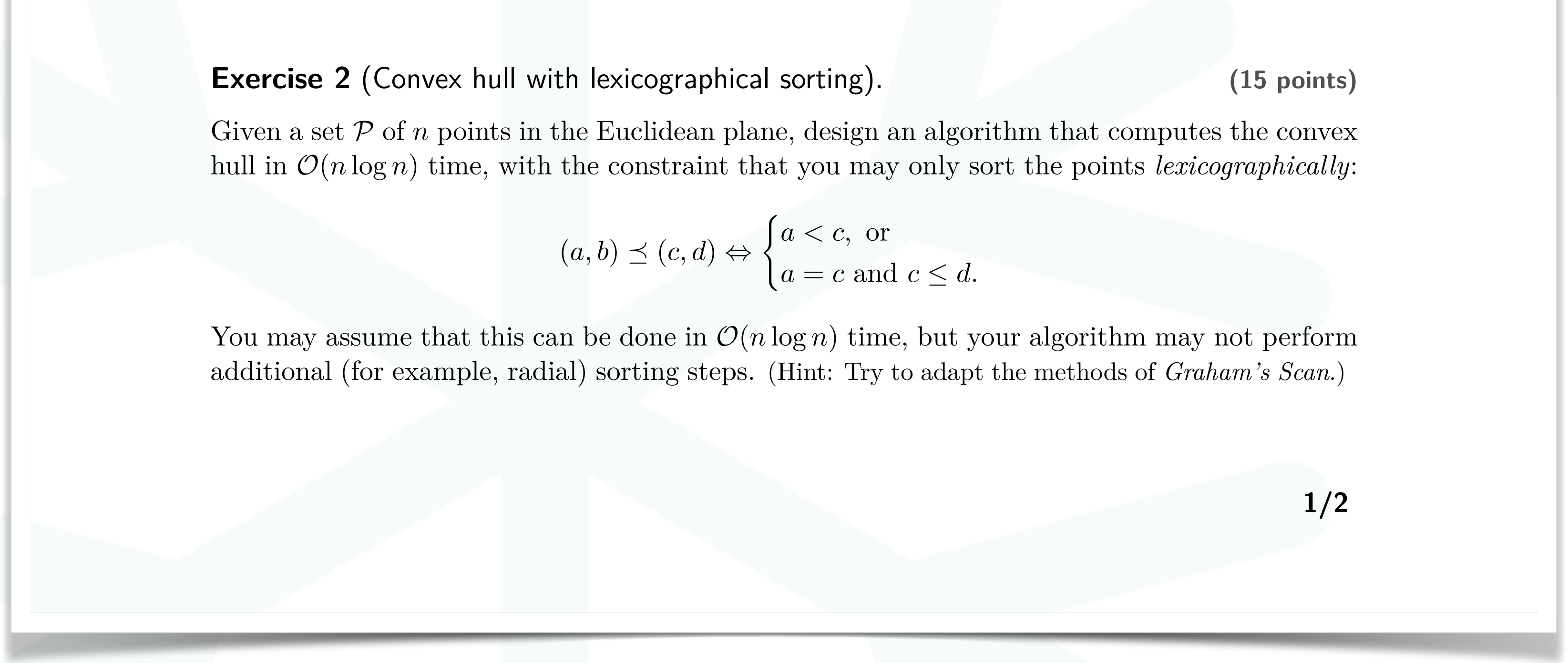


Figure 1: You require a location inside R that maximizes the distance to the nearest cookie outlet c_i.



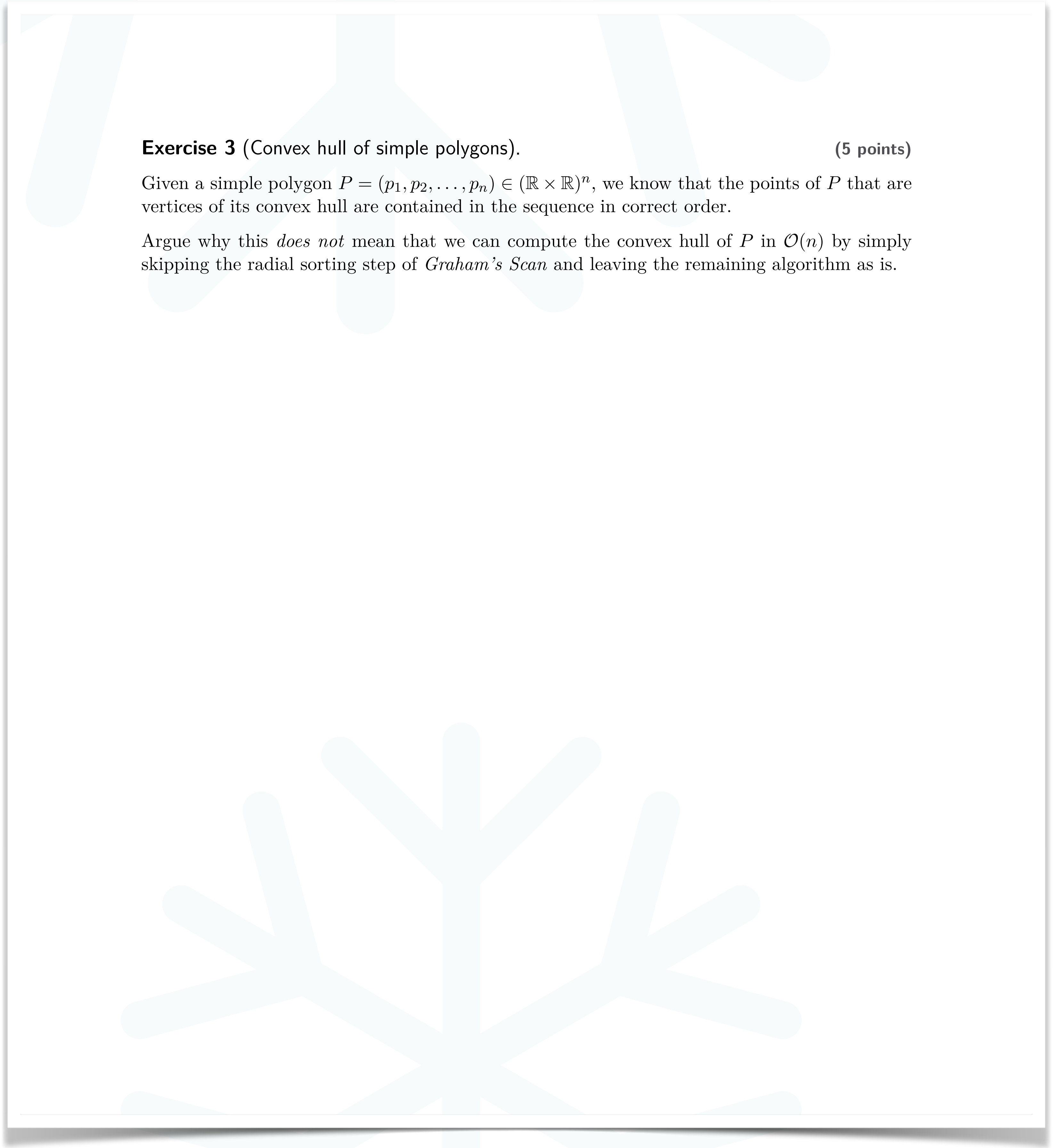
Exercise 2 (Convex hull with lexicographical sorting). **(15 points)**

Given a set $\mathcal{P}$ of n points in the Euclidean plane, design an algorithm that computes the convex hull in $\mathcal{O}(n \log n)$ time, with the constraint that you may only sort the points *lexicographically*:

$$(a,b) \preceq (c,d) \Leftrightarrow \begin{cases} a < c, \text{ or} \\ a = c \text{ and } c \leq d. \end{cases}$$

You may assume that this can be done in $\mathcal{O}(n \log n)$ time, but your algorithm may not perform additional (for example, radial) sorting steps. (Hint: Try to adapt the methods of *Graham's Scan*.)

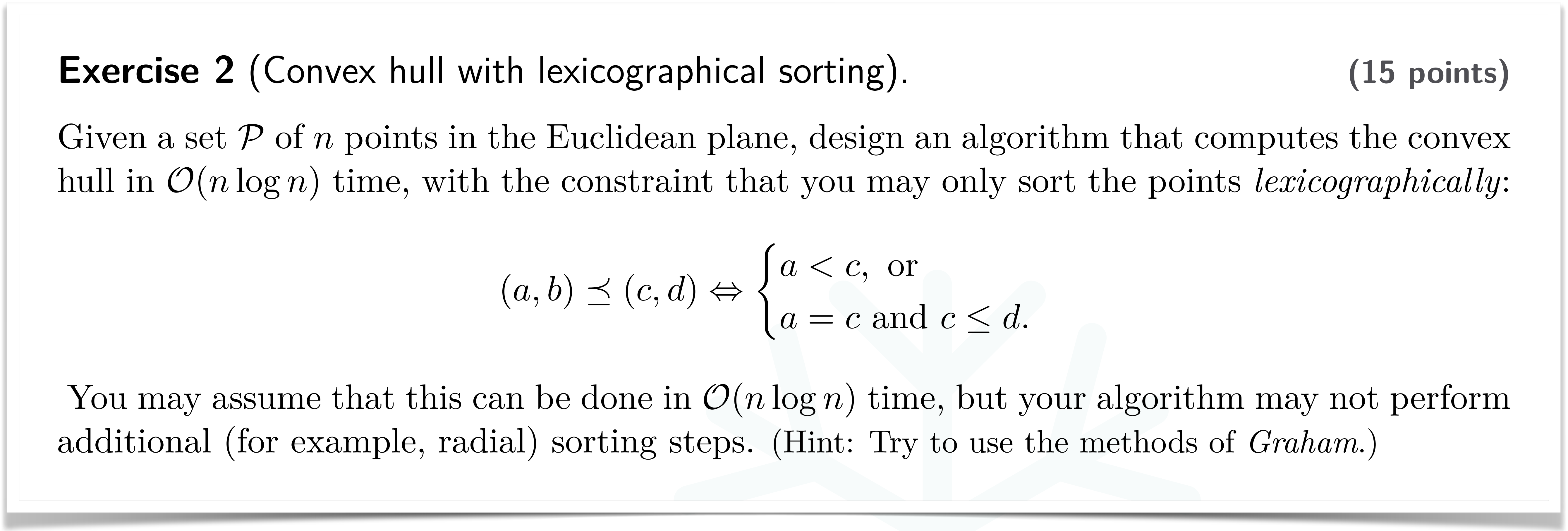
1/2



Exercise 3 (Convex hull of simple polygons). **(5 points)**

Given a simple polygon $P = (p_1, p_2, \ldots, p_n) \in (\mathbb{R} \times \mathbb{R})^n$, we know that the points of P that are vertices of its convex hull are contained in the sequence in correct order.

Argue why this *does not* mean that we can compute the convex hull of P in $\mathcal{O}(n)$ by simply skipping the radial sorting step of *Graham's Scan* and leaving the remaining algorithm as is.

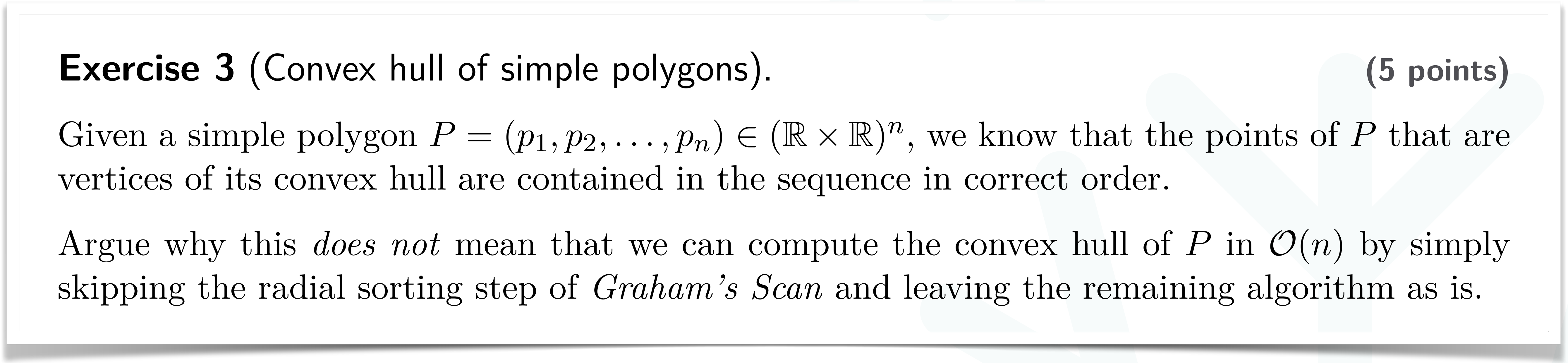


Exercise 2 (Convex hull with lexicographical sorting). **(15 points)**

Given a set $\mathcal{P}$ of n points in the Euclidean plane, design an algorithm that computes the convex hull in $\mathcal{O}(n \log n)$ time, with the constraint that you may only sort the points *lexicographically*:

$$(a,b) \preceq (c,d) \Leftrightarrow \begin{cases} a < c, \text{ or} \\ a = c \text{ and } c \leq d. \end{cases}$$

You may assume that this can be done in $\mathcal{O}(n \log n)$ time, but your algorithm may not perform additional (for example, radial) sorting steps. (Hint: Try to use the methods of *Graham*.)



Exercise 3 (Convex hull of simple polygons). **(5 points)**

Given a simple polygon $P = (p_1, p_2, \ldots, p_n) \in (\mathbb{R} \times \mathbb{R})^n$, we know that the points of P that are vertices of its convex hull are contained in the sequence in correct order.

Argue why this *does not* mean that we can compute the convex hull of P in $\mathcal{O}(n)$ by simply skipping the radial sorting step of *Graham's Scan* and leaving the remaining algorithm as is.