Self-approaching paths in simple polygons

Self-Approaching Paths in Simple Polygons*

Prosenjit Bose[1], Irina Kostitsyna[†2], and Stefan Langerman[3]

1 **School of Computer Science, Carleton University, Ottawa, Canada**
jit@scs.carleton.ca

2 **Computer Science Department, Université libre de Bruxelles (ULB), Brussels, Belgium**
irina.kostitsyna@ulb.ac.be

3 **Computer Science Department, Université libre de Bruxelles (ULB), Brussels, Belgium**
stefan.langerman@ulb.ac.be

Abstract

We study *self-approaching paths* that are contained in a simple polygon. A self-approaching path is a directed curve connecting two points such that the Euclidean distance between a point moving along the path and any future position does not increase, that is, for all points a, b, and c that appear in that order along the curve, $|ac| \geq |bc|$. We analyze the properties, and present a characterization of shortest self-approaching paths. In particular, we show that a shortest self-approaching path connecting two points inside a polygon can be forced to follow a general class of non-algebraic curves. While this makes it difficult to design an exact algorithm, we show how to find a self-approaching path inside a polygon connecting two points under a model of computation which assumes that we can calculate involute curves of high order.

Lastly, we provide an algorithm to test if a given simple polygon is self-approaching, that is, if there exists a self-approaching path for any two points inside the polygon.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases self-approaching path, simple polygon, shortest path, involute curve

Digital Object Identifier 10.4230/LIPIcs.SoCG.2017.21

1 Introduction

The problem of finding an optimal obstacle-avoiding path in a polygonal domain is one of the fundamental problems of computational geometry. Often a desired path has to conform to certain constraints. For example, a path may be required to be monotone [3], curvature-constrained [9], have no more than k links [15], etc. A natural requirement to consider is that a point moving along a desired path must be always getting closer to its destination. Such *radially monotone paths* appear, for example, in greedy geographic routing in a network setting [10], and beacon routing in a geometric setting [5]. A strengthening of a radially monotone path is a *self-approaching path* [12, 13, 1]: a point moving along a self-approaching path is always getting closer not only to its destination, but also to all the points on the path ahead of it. There are several reasons to prefer self-approaching paths over radially monotone paths. First, unlike for a radially monotone path, any subpath of a self-approaching path is self-approaching. Thus, if the destination is not known in advance and the desired path is required to be radially monotone, one would have to resort to using

[*] A full version of the paper is available at https://arxiv.org/abs/1703.06107.

[†] I. K. was supported in part by the NWO under project no. 612.001.106, and by F.R.S.-FNRS.

33rd International Symposium on Computational Geometry (SoCG 2017).
Editors: Boris Aronov and Matthew J. Katz; Article No. 21; pp. 21:1–21:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

self-approaching paths. Second, the length of a radially monotone path can be arbitrarily large in comparison with the Euclidean distance between the source and the destination points, whereas self-approaching paths have a bounded detour [13]. These properties make self-approaching paths of interest to various applications, including network routing, graph drawing, and others.

In this paper we study self-approaching paths that are contained in a simple polygon. We consider the following questions:

- Given two points s and t inside a simple polygon P, does there exist a self-approaching s-t path inside P?
- Find the shortest self-approaching s-t path.
- Given a point s in a simple polygon P, what is the set of all points reachable from s with self-approaching paths?
- Given a point t, what is the set of all points from which t is reachable with a self-approaching path?
- Given a polygon P, test if it is self-approaching, *i.e.*, if there exists a self-approaching path between any two points in P.

Related work. Self-approaching curves were first introduced in the context of online searching for the kernel of a polygon [12]. They were further studied in [13] where, among other results, the authors prove that the length of any self-approaching curve connecting two points is not greater than 5.3331 times the Euclidean distance between the points. An equivalent definition of a self-approaching path is that for every point on the path there has to be a 90° angle containing the rest of the path. Aichholzer *et al.* [1] developed a generalization of self-approaching paths for an arbitrarily fixed angle α. A relevant class of paths is increasing-chords paths [17], which are self-approaching in both directions. The nice properties of self-approaching and increasing-chords paths, and their potential to be applied in network routing, were recognized by the graph drawing community. As a result, a number of papers have appeared in recent years on self-approaching and increasing-chords graphs [2, 8, 16].

This paper is organized in the following way. We introduce a few definitions and concepts in Section 2. In Section 3, we characterize a shortest self-approaching path between two points in a simple polygon. In Section 4 we present an algorithm to construct the shortest self-approaching path between two points if it exists, or to report that it does not exist, by assuming a model of computation in which we can solve certain transcendental equations. Finally, in Section 5 we present a linear-time algorithm to decide if a polygon is self-approaching, that is, if there is a self-approaching path between any two point of the polygon. Due to space limitations, some proofs are omitted. For the details refer to the full version of this paper [6].

2 Preliminaries

For two points p_1 and p_2 on a directed path π that starts at point s, we shall say that $p_1 <_\pi p_2$ if p_1 lies between s and p_2 along π. For a directed path π and two points $p_1 <_\pi p_2$ on it, denote the subpath from p_1 to p_2 by $\pi(p_1, p_2)$.

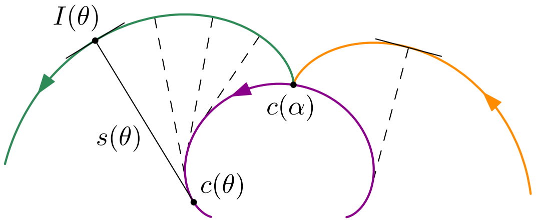

Figure 1 Curve $c(\theta)$ and two involutes. The arrows designate the direction of growth of the parameter θ. The involute on the left is defined by tangents pointing in the negative direction of c, and the involute on the right is defined by tangents pointing in the positive direction of c.

▶ **Definition 1.** A *self-approaching path* π in a continuous domain is a piece-wise smooth[1] oriented curve such that for any three points a, b, and c on it, such that $a <_\pi b <_\pi c$: $|ac| \geq |bc|$, where $|ac|$ and $|bc|$ are Euclidean distances.

Icking *et al.* [13] showed the following *normal property* of a self-approaching path, that we will be using extensively in this paper,

▶ **Lemma 2** (the normal property [13]). *An s-t path π is self-approaching if and only if any normal to π at any point $a \in \pi$ does not cross $\pi(a,t)$.*

▶ **Definition 3.** A normal h to a directed curve π at some point $a \in \pi$ defines two half-planes. Let the *positive half-plane* h^+ be the open half-plane which is congruent with the direction of π at point a.

We can rephrase the normal property in the following way.

▶ **Lemma 4** (the half-plane property). *An s-t path π is self-approaching if and only if, for any normal h to π at any point $a \in \pi$, the subpath $\pi(a,t)$ lies completely in the positive half-plane h^+.*

▶ **Definition 5.** A *bend* of a self-approaching path π is a point of discontinuity of the first derivative of π.

▶ **Definition 6.** A *reachable region* $\mathcal{R}(s) \subseteq P$, for a given point s in a polygon P, is the set of all points $t \in P$ for which there exists a self-approaching s-t path $\pi \in P$.

▶ **Definition 7.** A *reverse-reachable region* $\mathcal{R}^{-1}(t) \subseteq P$, for a given point t in a polygon P, is a set of all points $s \in P$ for which there exists a self-approaching s-t path $\pi \in P$.

2.1 Involutes

Next we introduce involute curves of kth order that will appear later as parts of shortest self-approaching paths.

An involute of a convex curve c is a curve traced by the end point of an unwinding pull-taut string rolled on c. Consider a parameterization $\vec{c}(\theta)$ of the curve, and let c be oriented in the direction of growth of the parameter θ. The involute of c can be computed by the following formula:

$$\vec{I}(\theta) = \vec{c}(\theta) - s(\theta)\frac{\vec{c}\,'(\theta)}{|\vec{c}\,'(\theta)|},$$

[1] Some previous works do not require the curve to be smooth. However in this paper we will be mostly considering shortest self-approaching paths, and thus the requirement on smoothness is justified.

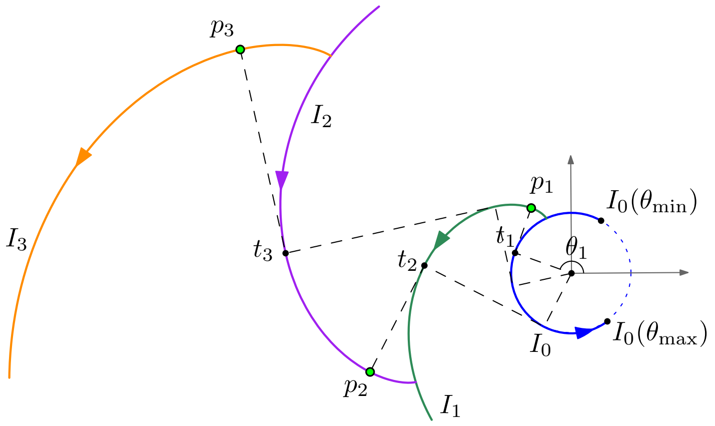

Figure 2 Circular arc $I_0(\theta)$, and three involutes $I_1(\theta)$, $I_2(\theta)$, and $I_3(\theta)$: for each i, $I_i(\theta)$ is an involute about $I_{i-1}(\theta)$ that passes through point p_i. The arrows designate the direction of growth of the parameter θ (they are not necessarily consistent with the direction of a self-approaching path).

where $s(\theta)$ is the length of the tangent segment $|\overline{c(\theta)I(\theta)}|$,

$$s(\theta) = \int\limits_{\alpha}^{\theta} |\vec{c}\,'(t)|dt\,.$$

The constant α defines the point at which the involute I will start unwinding around c (see Fig. 1). The involute has two branches: the *positive* branch unwinds starting at point α in the direction of growth of θ, and the *negative* branch unwinds in the opposite direction. If the curve c is defined on the interval $[\theta_{\min}, \theta_{max}]$, then the positive branch of its involute is defined on the interval $[\alpha, \theta_{\max}]$, and the negative – on the interval $[\theta_{\min}, \alpha]$.

We define an *involute of order* k of a curve $c(\theta)$ to be an involute of one branch (that contains the point corresponding to a parameter α_k) of an involute of order $k-1$ of $c(\theta)$, with an involute of order 0 being the curve $c(\theta)$ itself,

$$\vec{I}_k(\theta) = \vec{I}_{k-1}(\theta) - s_k(\theta)\frac{\vec{I}'_{k-1}(\theta)}{|\vec{I}'_{k-1}(\theta)|}\,, \qquad \text{where} \qquad s_k(\theta) = \int\limits_{\alpha_k}^{\theta} |\vec{I}'_{k-1}(t)|dt\,,$$
$$\vec{I}_0(\theta) = \vec{c}(\theta)\,.$$

In the following sections we will show that shortest self-approaching paths consist of straight-line segments, circular arcs, and involutes of circular arcs of some order. In the full version of this paper [6] we provide the details of the derivation of the following formula for an involute of a circle of order k:

$$I_k(\theta) = \sum_{0}^{\lfloor \frac{k}{2} \rfloor} (-1)^i a_{2i}(\theta) \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} - \sum_{0}^{\lceil \frac{k}{2} \rceil - 1} (-1)^i a_{2i+1}(\theta) \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix}\,,$$

where each involute I_i passes through a point $p_i(r_i, \varphi_i)$ for all $1 \le i \le k$,

$$a_i(\theta) = r_0\frac{\theta^i}{i!} + c_1\frac{\theta^{i-1}}{(i-1)!} + \cdots + c_i\,,$$

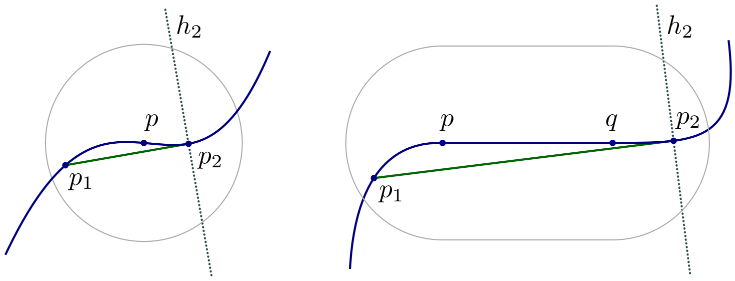

Figure 3 If a self-approaching path has an inflection point (or a segment) interior to P, then there exists a shortcut.

and the constants c_i can be found from the following equations:

$$r_i \cos(\theta_i - \varphi_i) = \sum_{0}^{\lfloor \frac{i}{2} \rfloor} (-1)^j a_{2j}(\theta_i), \qquad r_i \sin(\theta_i - \varphi_i) = \sum_{0}^{\lceil \frac{i}{2} \rceil - 1} (-1)^j a_{2j+1}(\theta_i)\,. \qquad (1)$$

The length $|\overline{p_k t_k}|$ of the tangent segment equals $|a_k(\theta_k)|$.

3 Properties of a shortest self-approaching path

In this section we will prove the following properties of a shortest self-approaching path from s to t inside a simple polygon P:

- A shortest self-approaching path is unique.
- The shortest self-approaching path consists of straight segments, circular arcs and involutes to the latter pieces of the path.

We begin by proving several lemmas:

▶ **Lemma 8.** *For any two points $p_1 <_\pi p_2$ on a self-approaching s-t path π in $\mathbb{R}^2$, the perpendicular bisector of the straight-line segment $\overline{p_1 p_2}$ does not intersect the subpath $\pi(p_2, t)$.*

Proof. Let h^- be the half-plane defined by the perpendicular bisector of segment $\overline{p_1 p_2}$ that contains p_1. Assume there is a point q on the subpath $\pi(p_2, t)$ that is interior to h^-. Then $|p_1 q| < |p_2 q|$, which contradicts the definition of a self-approaching path. ◀

▶ **Lemma 9.** *Bends of a shortest self-approaching path in a simple polygon P form a subset of vertices of P.*

Thus, any point of a shortest self-approaching s-t path which is interior to P has a well-defined tangent. This point is an *inflection point*, if its tangent separates the self-approaching path in a small enough ε-neighborhood. We can also introduce a notion of an *inflection segment* for a path that contains a straight-line segment as a subpath. A straight-line segment of a path is an inflection segment if its supporting line separates the path in a small enough ε-neighborhood around the segment (refer to Fig. 3).

▶ **Lemma 10.** *A shortest self-approaching s-t path in a simple polygon P cannot have an inflection point (or an inflection segment) that is interior to P.*

Proof. Suppose a shortest self-approaching s-t path π has an inflection point p (or an inflection segment $\overline{pq}$) interior to P. Consider an ε-neighborhood of p (or $\overline{pq}$) for some small ε such that it is also interior to P, and it does not contain other inflection points. Choose a

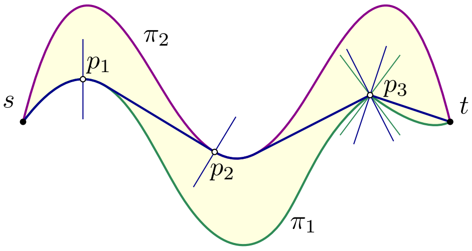

Figure 4 A geodesic bounded between two self-approaching s-t paths is also self-approaching.

point p_1 on subpath $\pi(s,p)$ close to p and draw a tangent through it to a subpath of $\pi(p,t)$ contained in the ε-neighborhood (refer to Fig. 3). Let p_2 be the tangent point. We can always choose p_1 such that the segment $\overline{p_1p_2}$ lies inside the ε-neighborhood. Let h_2 be the normal line to π drawn through p_2. Because π is self-approaching, the subpath $\pi(p_2,t)$ lies in the positive half-plane h_2^+. Therefore, none of the normal lines to $\overline{p_1p_2}$ intersects the subpath $\pi(p_2,t)$. Thus, $\pi(s,p_1) \oplus \overline{p_1p_2} \oplus \pi(p_2,t)$ is self-approaching and is shorter than π. ◀

Define the *inflection* points of a directed geodesic path γ from s to t as the first points of the inflection segments of γ, *i.e.*, the set of last points in the maximal subchains of γ with the same direction of turn.

▶ **Lemma 11.** *A shortest self-approaching path from s to t in a simple polygon P contains all the inflection points of the geodesic path from s to t.*

Proof. Consider an inflection segment $\overline{p_ip_j}$ of the geodesic path γ from s to t, p_i is one of its inflection points. Any shortest self-approaching path π intersects $\overline{p_ip_j}$. If the intersection point were not p_i, then π would contain an inflection point that is interior to P, but this would contradict Lemma 10. ◀

Consider two self-approaching paths π_1 and π_2 in a simple polygon P from s to t that do not have other points in common. Let γ be a geodesic path from s to t inside the area bounded by π_1 and π_2. Then, the following lemma holds.

▶ **Lemma 12.** *A geodesic path γ between two self-approaching paths π_1 and π_2 is also self-approaching.*

Proof. We use the fact that the geodesic lies inside of the convex hull of each side of the boundaries between which it is constrained, *i.e.*, $\gamma \subset CH(\pi_1)$ and $\gamma \subset CH(\pi_2)$.

Any point $p \in \gamma$ either lies on one of the paths π_1 and π_2 or on a straight line segment that is bitangent to the boundary (refer to Fig. 4).

Consider the case when p lies on π_1 or π_2, and is not a bend point (as point p_1 in the figure). Let, w.l.o.g., $p \in \pi_1$. The positive half-plane h^+ of the normal to π_1 at p contains the rest of the path $\pi_1(p,t)$. Therefore it contains the convex hull of $\pi_1(p,t)$, and the subpath $\gamma(p,t)$ of the geodesic.

When p lies on a path π_1 and is a bend point, the two normals to the path at p define two positive half-planes whose intersection contains the rest of the path from p to t. The two normals to the geodesic path at this point will lie in between the two normals to the boundary path (as in the figure for point p_3). Thus, the intersection of the two positive half-planes of the normals to the geodesic contains the convex hull of the subpath from s to t, and, therefore, the rest of the geodesic path $\gamma(p,t)$.

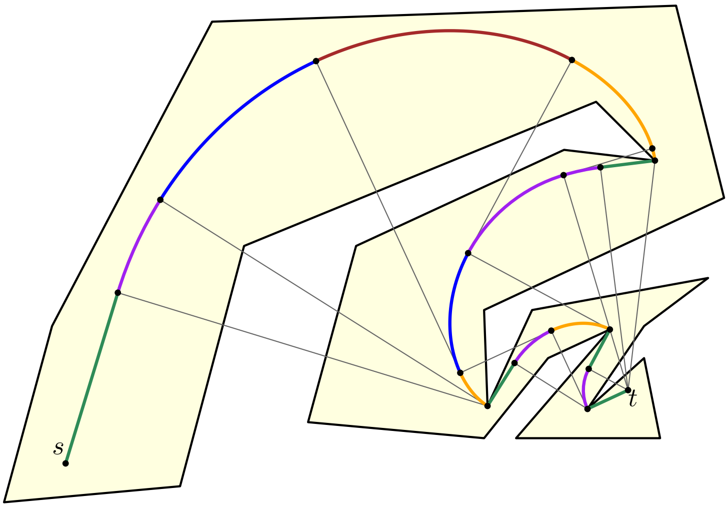

Figure 5 Shortest self-approaching path from s to t consists of straight-line segments, circular arcs, and involutes of a circle of some order. Straight segments are shown in green, circular arcs in purple, involutes of a circle of first order in orange, involutes of a circle of a second order in blue, and involutes of a circle of third order in brown.

In the case when p lies on a bitangent, consider its end point p_2. The normal to γ at p is parallel to the normal to γ at p_2. By one of the cases considered above, the positive half-plane at p_2 (or the intersection of two positive half-planes) will contain $\gamma(p_2, t)$, and, therefore, the positive half-plane of the normal to γ at p will contain the subpath $\gamma(p, t)$.

Thus, by the half-plane property, γ is self-approaching. ◀

As a corollary to this lemma, for two self-approaching paths from s to t, a path, composed of geodesics in the areas bounded by subpaths of the two paths between each pair of consecutive intersection points, is also self-approaching. In other words, let $s = p_0, p_1, \ldots, p_k, p_{k+1} = t$ be all the intersection points of π_1 and π_2 in the order they appear on π_1 and π_2. Observe that the intersection points must appear in the same order along the both paths, otherwise there would exist three points on one of these paths which would violate the definition of a self-approaching path. Let γ_i be the geodesic from p_i to p_{i+1} in the area between the two subpaths $\pi_1(p_i, p_{i+1})$ and $\pi_2(p_i, p_{i+1})$. Then,

▶ **Lemma 13.** *The concatenation of the geodesics $\gamma = \gamma_0 \oplus \gamma_1 \oplus \cdots \oplus \gamma_k$ is self-approaching.*

Proof. By a similar argument as in Lemma 12, for any normal to γ_i at point p, its positive half-plane either contains the convex hull of $\pi_1(p, t)$, or it contains the convex hull of $\pi_2(p, t)$. In both cases, that implies that the subpath $\gamma(p, t)$ lies in the positive half-plane of the normal. Therefore, γ is self-approaching. ◀

The next theorem is a direct corollary of Lemma 13.

▶ **Theorem 14.** *A shortest self-approaching s-t path is unique.*

Figure 5 shows an example of a shortest self-approaching path inside a polygon. In the following theorem we give its characterization.

▶ **Theorem 15.** *A shortest self-approaching s-t path in a simple polygon consists of straight-line segments, circular arcs, and circle involutes of some order.*

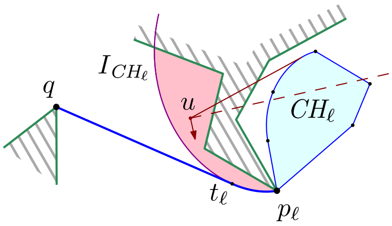

Figure 6 Illustration to Theorem 15.

Proof. Let $p_1, p_2, \ldots, p_k$ be the points of the shortest self-approaching s-t path π^* in the order from s to t, in which the path touches the boundary of P. Consider the last segment $\pi^*(p_k, t)$. It is a straight-line segment. Otherwise it could be shortened in the following way. Consider the last segment $\overline{qt}$ of a geodesic path from s to t, and extend it in the direction from t to q until intersecting path π^*; denote the intersection point as q'. (Note, that it exists, as the extension of $\overline{qt}$ beyond q until intersecting the boundary of P separates s from t.) Then, π^* can be shortened by replacing $\pi^*(q', t)$ by the segment $\overline{q't}$.

Now, suppose that all the segments $\pi^*(p_i, p_{i+1})$ consist of straight-line segments, circular arcs, or involutes of a circle of some order for all $i > \ell$ for some ℓ. We will show, that then, the segment $\pi^*(p_{\ell-1}, p_\ell)$ consists of straight-line segments, circular arcs, and/or involutes.

Denote $CH_\ell = CH(\pi^*(p_\ell, t))$. Let, w.l.o.g., π^* touch the boundary of the polygon at point p_ℓ on its left side (refer to Fig. 6). Then construct an involute I_{CH_ℓ} of the convex hull CH_ℓ starting at point p_ℓ with the tangent point moving in clockwise direction around CH_ℓ until the first intersection point of the involute with the boundary of P. The area D_ℓ on the concave side of the involute that it cuts off of the polygon P is a "dead" region for any self-approaching path that ends with the subpath $\pi^*(p_{\ell,t})$ (red area in the Fig. 6). In other words, for any point $u \in D_\ell$, any path connecting u to p_ℓ will have a normal that intersects CH_ℓ, and therefore the subpath $\pi^*(p_\ell, t)$. To show that, consider any piecewise-smooth path π_u from u to p_ℓ. Parameterize π_u for some parameter $\tau \in [0, 1]$, where $\pi_u(0) = u$ and $\pi_u(1) = p_\ell$. Consider the distance function $d_u(\tau)$ from a point moving along π_u to the involute I_{CH_ℓ}. This function will be piecewise smooth as both of the paths are piecewise-smooth. As a point, moving along π_u, has to eventually coincide with p_ℓ, there exists parameter τ' at which the distance function is decreasing, and therefore, the angle between a tangent vector to π_u at the point $u' = \pi_u(\tau')$ and a tangent from u' to the convex hull CH_ℓ is greater than $90°$. Therefore, a positive half-plane of the normal to π_u at point u' does not fully contain the convex hull CH_ℓ, and therefore, the path $\pi_u \oplus \pi^*(p_\ell, t)$ is not self-approaching.

Now, consider a geodesic path from s to p_ℓ in the region $P \backslash D_\ell$, and consider its last segment qp_ℓ, where q is the last point before p_ℓ that belongs to the boundary of P. This segment can be a straight-line segment, or a straight-line segment $\overline{qt_\ell}$ followed by a piece of the involute I_{CH_ℓ}, where $\overline{qt_\ell}$ is tangent to I_{CH_ℓ}. If segment qp_ℓ is not on π^*, then, by a similar argument as above, we can show that π^* can be shortened. Extend the segment $\overline{qt_\ell}$ beyond the point q until the intersection q' with π^*. Then, π^* can be shortened if the subpath $\pi^*(q'p_\ell)$ is replaced by the segment qp_ℓ of the geodesic.

The boundary of the convex hull CH_ℓ consists of straight-line segments and pieces of the subpath $\pi^*(p_\ell, t)$, which we assumed were straight segments, arcs, and circle involutes. Therefore, the segment qp_ℓ of the geodesic path also consists of straight segments, circular arcs, and circle involutes, possibly, of one order higher than the following subpath. Therefore, the shortest self-approaching path consists of straight-line segments, circular arcs, and circle involutes of some order, that is not higher than the number of bends on the path. ◀

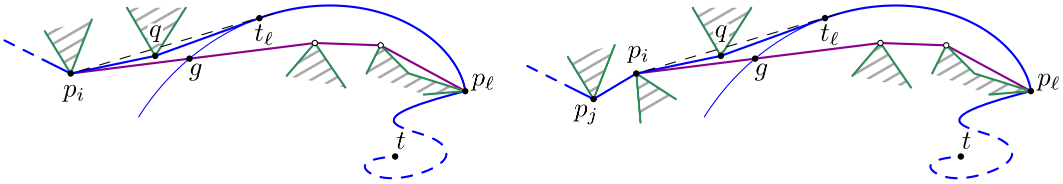

Figure 7 A subpath of a shortest self-approaching s-t path π^* (in blue) between two consecutive inflection points of the geodesic path γ (in purple) from s to t is geodesically convex. The last bend q of π^* before the vertex p_ℓ does not necessarily belong to γ.

In the last proof, the point q of the last segment qp_ℓ of the geodesic path from s to p_ℓ in $P \backslash D_\ell$ does not necessarily belong to the geodesic path from s to t. Consider an example in Fig. 7. In it, several vertices of the geodesic path γ are in the dead region (on the concave side of the involute). The tangent line from the last vertex (p_i in the left example, and p_j in the right example) of γ before p_ℓ that is not in the dead region intersects the boundary of the polygon. Angle $\angle p_i g t_\ell$, where g is the intersection point of γ with the involute, is an obtuse angle. This follows from the fact that the intersection angle between the straight-line segment $\overline{p_i p_\ell}$ and the tangent to the involute at the intersection point must not be greater than 90°, otherwise the point p_ℓ would not lie in the positive half-plane of the normal to the involute at the intersection point. Then, the total turn angle of the self-approaching path π^* from p_i to t_ℓ is less than 90°, and thus, the subpath $\pi^*(p_i, t_\ell)$ consists of straight-line segments. Let the previous inflection point of γ before p_ℓ be p_j, and the next inflection point of γ on or after p_ℓ be p_k. It follows then that the subpath $\pi^*(p_j, p_k)$ is *geodesically convex*, that is, the shortest path between any two points on $\pi^*(p_j, p_k)$ lies completely on one (and the same side) of the path. We obtain the following lemma.

▶ **Lemma 16.** *A shortest self-approaching s-t path in a simple polygon P consists of geodesically convex paths between inflection points of the geodesic from s to t.*

▶ **Theorem 17.** *A shortest self-approaching s-t path in a simple polygon P with n vertices consists of $O(n^2)$ segments. There exists a simple polygon P and two points s and t in it, such that the shortest self-approaching from s to t has $\Omega(n^2)$ segments.*

4 Existence of a self-approaching path

In this section we consider the question of testing whether, for given points s and t in a polygon P, they can be connected with a self-approaching path. In Theorem 15 we proved that a shortest self-approaching path can consist of involutes of a circle of high order, and in Section 2 we showed that such an involute is defined by a system of transcendental equations. In [14] Laczkovich proved a strengthening of Richardson's theorem, which states that in general the statement $\exists x : f(x) = 0$ is undecidable, where $f(x)$ is an expression generated by the rational numbers, the variable x, the operations of addition, multiplication, composition, and the sine function. Equations (1) describing the involutes are a special case of the class of expressions in Laczkovich's theorem. Nevertheless, it strongly suggests that an involute of a circle of order higher than one cannot be computed.

Next, we show an algorithm to test whether there exists a self-approaching path connecting two points s and t, and if so, to compute the shortest path, under the assumption that we can solve Equations (1). Subsequently, it is possible to release this assumption, and modify the algorithm to build an approximate solution, given that the shortest self-approaching path from s-to-t exists and there is a small leeway around it free of the polygon boundary points.

4.1 Shortest path algorithm

The proof of Theorem 15 is constructive. Assume that we can solve equations of the form as Equations (1) for an involute of order k in time $O(f(k))$, and evaluate the formula of the involute of order k for a given parameter θ in time $O(g(k))$. Then, we can decide if two points s and t can be connected by a self-approaching path, and we can construct the shortest path between the points. The outline of the algorithm is:

- Starting at t, move backwards along a geodesic s-t path γ. Maintain the convex hull CH of the final part of the shortest self-approaching path π^* to the destination t built so far.
- At every bend point p_ℓ:
  - Calculate the appropriate branch of an involute I_{CH} of the convex hull CH. If I_{CH} intersects the opposite boundary of the polygon, thus separating s from t, report that a self-approaching path from s to t does not exist and terminate the algorithm.
  - Otherwise, find a geodesic path γ_ℓ from the preceding inflection point of γ to p_ℓ in $P \backslash I_{CH}$, and add its last segment qp_ℓ as a prefix to π^*.
  - Update the convex hull CH. Repeat for the new bend point q, until s is reached. Report the found path π^*.

To obtain an algorithm with an optimal running time, there are a few considerations to take into account when constructing the shortest path. First, instead of unnecessarily constructing the whole involute I_{CH} until the intersection point with the boundary of P, and then discarding the part of it under the tangent line from q, its segments can be built one by one as needed up to the tangent point. Second, to optimally test if I_{CH} intersects the opposite boundary of the polygon, we can maintain a shortest path tree that will allow us to build funnels from the opposite sides of the polygon boundary. Third, it is not necessary to construct the whole geodesic γ_ℓ to be able to compute its last segment qp_ℓ. Instead, we can move backwards along γ, vertex by vertex, until we reach a point from which the tangent to I_{CH} can be constructed (possibly with adding new points along it).

Let the edges of P be oriented in counter-clockwise order. We shall call the two ends of an edge e, the *front*-point, and the *end*-point.

Next, we present the details of the algorithm.

Initialization step. Compute the shortest path tree SPT_s with root s [11], and preprocess it to answer the lowest common ancestor query [4]. Compute the geodesic γ from s to t, and store γ as a stack of vertices. Let the first and the last segments of γ be $\overline{sp'}$ and $\overline{p''t}$ respectively. Extend $\overline{sp'}$ beyond s until intersection with ∂P at some point a, and extend $\overline{p''t}$ beyond t until intersection with ∂P at some point b. This can be done in $O(\log n)$ time with a ray-shooting query after linear-time preprocessing of the polygon [7]. Let L be the chain of the boundary of P from b to a in counter-clockwise order, we shall call it the *left* chain. Similarly, let the *right* chain R be the chain of the boundary of P from a to b in counter-clockwise order. Initialize π^* and CH with the last segment $\overline{p''t}$ of γ, and pop the point t from γ.

The main loop. Let p_ℓ be the point on top of the stack γ, before the beginning of the current iteration of the loop. Let π^* touch ∂P in point p_ℓ on its left side (the case when π^* touches ∂P on its right side is equivalent). Let CH be the convex hull of the already built subpath $\pi^*(p_\ell, t)$.

Pop p_ℓ from the top of the stack γ. Consider the previous segment $\overline{p_i p_\ell}$ of γ (point p_i is currently on top of the stack γ).

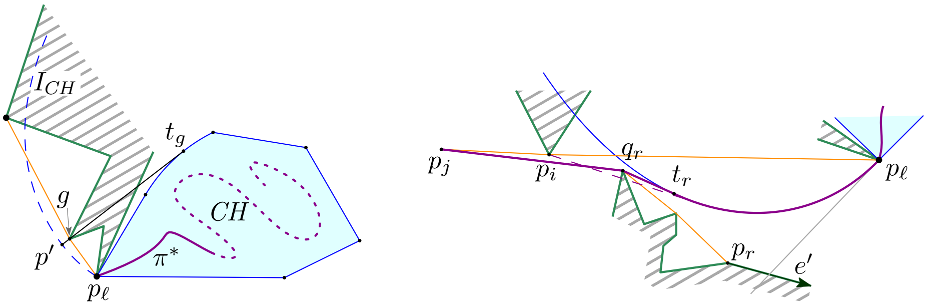

Figure 8 Illustration for Case 2 of the algorithm.

Case 1. If the angle between $\overline{p_\ell p_i}$ and the tangent in the clockwise direction to CH at point p_ℓ form an angle that is not less than 90°, then $\overline{p_i p_\ell}$ lies on the shortest self-approaching path from s to t; append $\pi^*(p_\ell, t)$ with $\overline{p_i p_\ell}$ in the front, and update the convex hull.

Case 2. If the angle between $\overline{p_\ell p_i}$ and the tangent in the clockwise direction to CH form an angle that is less than 90°, we need to calculate the involute I_{CH} of the convex hull for the tangent point moving clockwise around the boundary of CH starting at p_ℓ. We first will determine until which point to calculate I_{CH}.

First, we check whether the points on the geodesic path γ before p_ℓ lie in the dead region defined by I_{CH}. To do that without explicitly constructing I_{CH} first, for each point g on top of the stack γ, we construct a tangent line to CH which is leaving it on its right side (refer to Fig. 8 (left)). Let t_g be the tangent point on CH, and let $t_g \in I_g$ for some involute segment I_g on the boundary of the convex hull. Let $\overline{gt_g}$ intersect I_{CH} at point p'. We know that the length of the segment $\overline{p't_g}$ is equal to the length of the boundary of the convex hull CH from t_g to p_ℓ. Thus, to check whether g lies in the dead region we can compare the length of the segment $\overline{gt_g}$ to the length of the boundary of CH from t_g to g. If g does lie in the dead region, we simply remove it from the top of the stack γ, and proceed. If at some moment γ becomes empty, *i.e.*, the point s lies in the dead region, we report that a self-approaching path from s to t does not exist and terminate the algorithm.

Now, let p_i be the first point on γ before p_ℓ that does not lie in the dead region of I_{CH}. As in Fig. 7, the tangent segment from p_i to the involute may intersect the right chain of the boundary of P. Moreover, the right chain of the boundary of P may intersect I_{CH}. To test and account for that case, we do the following. Let $\vec{\tau} = \vec{I}'_{CH}(p_\ell)$ be the tangent vector to I_{CH} at point p_ℓ. Run a ray shooting query from p_ℓ in the direction $-\vec{\tau}$. Let it intersect an edge e' of R, and denote its front-point as p_r (refer to Fig. 8 (right)). Then, find a vertex p_j in the shortest path tree SPT_s that is the lowest common ancestor of p_ℓ and p_r. Let γ_ℓ and γ_r be the two shortest paths from p_j to p_ℓ and to p_r respectively. Paths γ_ℓ and γ_r form two convex chains. If γ_r does not intersect I_{CH}, then either a common tangent to γ_ℓ and I_{CH}, or a common tangent to γ_r and I_{CH}, will belong to π^*. To be able to compute the common tangents, we now explicitly construct I_{CH} segment by segment until a certain point. Let $p'p''$ be the last segment of I_{CH} constructed so far (with the curve orientation from p' to p''). We stop the construction of I_{CH} when the segment $\overline{p'p_j}$ makes a left turn with respect to the tangent vector $-\vec{\tau}$, where $\vec{\tau} = \vec{I}'_{CH}(p')$.

Whether γ_r intersects I_{CH} can be found during the computation of the common tangent. If it does, report that s and t cannot be connected with a self-approaching path and terminate the algorithm.

Let q_ℓ and q_r be the two tangent points on γ_ℓ and γ_r respectively of the common tangent lines with I_{CH}. One of the points q_ℓ and q_r, or both, will be equal to p_j.

- If $p_j = q_\ell = q_r$, then append π^* with $\overline{p_j t_j} \oplus I_{CH}(t_j, p_\ell)$, where t_j is the tangent point on I_{CH}.
- If $p_j = q_r \neq q_\ell$, then append π^* with $\gamma(p_j, q_\ell) \oplus \overline{q_\ell t_\ell} \oplus I_{CH}(t_\ell, p_\ell)$, where t_ℓ is the tangent point on I_{CH} of the common tangent with γ_ℓ.
- If $p_j = q_\ell \neq q_r$, then append π^* with $\gamma(p_j, q_r) \oplus \overline{q_r t_r} \oplus I_{CH}(t_r, p_\ell)$, where t_r is the tangent point on I_{CH} of the common tangent with γ_r.

Remove the points from γ until p_j is on top of the stack, and update CH. Iterate over the main loop until γ is empty, and return π^*.

In the full version of this paper [6] we discuss how to compute common tangents between a chain of involute segments of order $\leq k$ of size n and a polygonal chain of size m in $O(\log(m+n) + g(k) \log m + f(k))$ time, and between two chains of involutes of sizes n and m in $O(\log(m+n) + g(k) \log m + f(k))$ time. We also show how to test if two chains intersect in $O(\log(m+n) + (g(k) + f(k)) \log m)$ time.

Maintaining CH. At the end of each iteration of the main algorithm, we need to update the convex hull of the subpath of the shortest self-approaching path built so far. This can involve finding a tangent from a point to a chain of involutes, or finding a common tangent of two chains of involutes.

Moreover, we want to be able to optimally calculate the length of a boundary from the current point p_ℓ to some point t_g. For that, associate two values $\text{dist}_{cw}(u)$ and $\text{dist}_{ccw}(u)$ to each end point of a segment on CH that will contain the distance to p_ℓ (up to some constant that will be equal for all the points) along the boundary in clockwise and counter-clockwise direction, respectively. Moreover, for two points u and v on CH, the length of the boundary between them can be calculated by $\text{dist}_{ccw}(v) - \text{dist}_{ccw}(u)$, if the chain of CH between u and v in counter-clockwise order does not contain p_ℓ. This fact will allow us to maintain the values in the points unchanged when updating the convex hull.

At every iteration of the algorithm, the distance from some tangent point t_g on an involute segment $p'p''$ to p_ℓ in the clockwise direction can be computed by formula $s(t_g) = \text{length}_I(t_g, p'') + \text{dist}_{cw}(p'') - \text{dist}_{cw}(p_\ell)$, where $\text{length}_I(t_g, p'')$ is the arc length of the involute from point t_g to p''. Analogously, the distance from t_g to p_ℓ in the counter-clockwise direction can be computed by taking $s(t_g) = \text{length}_I(t_g, p') + \text{dist}_{ccw}(p') - \text{dist}_{ccw}(p_\ell)$.

When updating the convex hull after extending the path π^*, we calculate the lengths of the tangent segments and the new involute arcs, and set the values $\text{dist}_{cw}(u)$ and $\text{dist}_{ccw}(u)$ to the new points of CH relatively to the values of the points remaining on CH. This will take $f(k)$ time to compute the arc length per segment of an involute of order k.

Taking these considerations into account, we conclude with the following theorem:

▶ **Theorem 18.** *The algorithm above constructs a shortest self-approaching path from s to t or reports that it does not exist in $O(K + \frac{n \log K}{\sqrt{K}}(g(\sqrt{K}) + f(\sqrt{K})))$ running time, where K is the size of the output, $f(k)$ is the time it takes to compute an involute of order k, and $g(k)$ is the time it takes to evaluate an involute of order k at a given point.*

5 Self-approaching polygon

A polygon is self-approaching, if for any two points there exists a self-approaching path connecting them.

▶ **Theorem 19.** *Polygon P is self-approaching if and only if for any disk D centered at any point $p \in P$, the intersection $D \cap P$ has one connected component.*

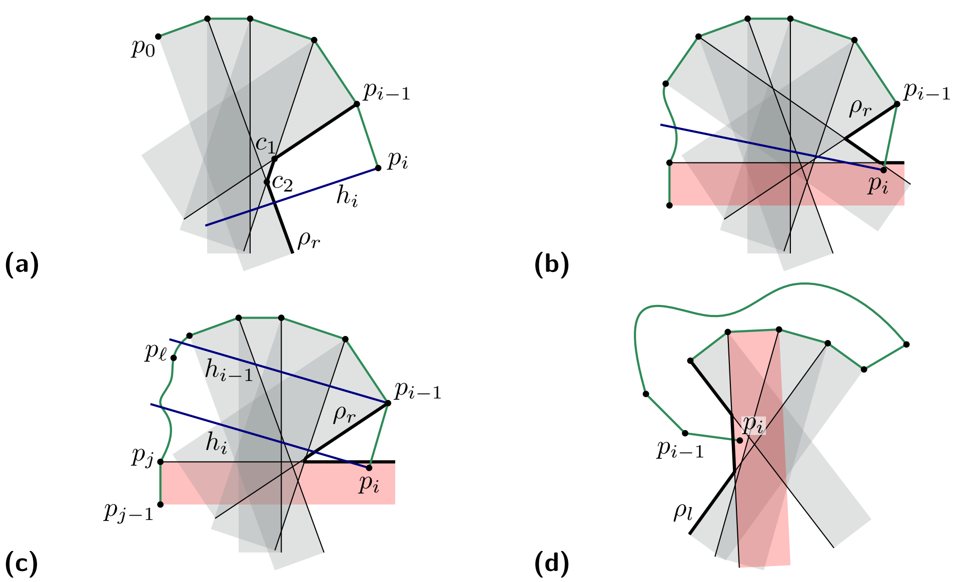

Figure 9 The illustration for Theorem 21.

Recall that a path is increasing-chord if it is self-approaching in both directions.

▶ **Corollary 20.** *Any self-approaching polygon is also increasing-chord.*

Next, we present an algorithm to test whether a given simple polygon P is self-approaching. Observe that from the proof of Theorem 19 the following property holds: the polygon P is self-approaching if and only if an area bounded between the two normals to e at its two end points in the right half-plane of e is free of ∂P, for all edges e on the boundary of P directed in counter-clockwise order. We call this area the *half-strip* of e. We will use this property to test efficiently if the polygon is self-approaching.

Let P be given as a set of points $p_0, p_1, \dots, p_{n-1}$ in counter-clockwise order around the boundary. We will start at p_0, move along the boundary in counter-clockwise order and maintain the union of all the half-strips of the edges visited so far. More precisely, we will maintain the left and the right sides, ρ_l and ρ_r, of the hour-glass shape that is the union of the half-strips; ρ_l and ρ_r are convex polygonal chains (refer to Fig. 9). Store the segments of ρ_l and ρ_r as two lists, the last segments in the lists are infinite rays.

At every iteration of the algorithm, perform the following steps. Let p_i be the current point of the polygon P. The chain ρ_r contains the right side of the union of all the half-strips up to point p_{i-1}. Consider the next boundary segment $\overline{p_{i-1}p_i}$, and a perpendicular ray h_i at the point p_i (refer to Fig. 9 (a)). To update the chain ρ_r, do the following: Traverse ρ_r, and for every its segment $\overline{c_jc_{j+1}}$,

- if $\overline{p_{i-1}p_i}$ intersects $\overline{c_jc_{j+1}}$, then report that P is not self-approaching and terminate;
- if h_i intersects $\overline{c_jc_{j+1}}$, calculate the intersection point c', and replace the first elements of the list ρ_r up to $\overline{c_jc_{j+1}}$ with two segments, $\overline{p_ic'}$ and $\overline{c'c_{j+1}}$; repeat for the next point p_{i+1}.

Traverse the boundary of polygon P twice in counter-clockwise order, and then repeat the same algorithm traversing the boundary of P twice in clockwise order. If none of the segments $\overline{p_{i-1}p_i}$ intersected a segment of ρ_r, report that P is self-approaching.

▶ **Theorem 21.** *Given a simple polygon P with n vertices, the presented algorithm tests in $O(n)$ time if it is self-approaching.*

Proof. Consider the counter-clockwise traversal of the boundary of P. There are two cases when the boundary segment $\overline{p_{i-1}p_i}$ intersects ρ_r. In the first case, $\overline{p_{i-1}p_i}$ intersects ρ_r, and h_i does not intersect it (refer to Fig. 9 (b)). Let us call it the intersection of type 1. In the second case, $\overline{p_{i-1}p_i}$ intersects ρ_r after h_i intersects it (refer to Fig. 9 (c)). Let us call it the intersection of type 2. Moreover, the boundary segment $\overline{p_{i-1}p_i}$ may intersect ρ_l (refer to Fig. 9 (d)). Let us call it the intersection of type 3.

When traversing the polygon counter-clockwise, the presented algorithm will recognize the first type of the intersection, but not the second or third type. In case of the second type, during one iteration, the algorithm stops traversing ρ_r after finding the intersection point of h_i, and thus will not find the intersection of the segment with ρ_r. And in case of the third type, the algorithm does not check for intersection with ρ_l at all.

Nevertheless, we will prove, that by repeating the checks above twice and in two directions, counter-clockwise from p_0 to p_{n-1}, and clockwise from p_{n-1} to p_0, the algorithm will correctly decide if the polygon is self-approaching or not.

Case 0. If the polygon is self-approaching, then none of the segments will intersect ρ_r or ρ_l. The algorithm will traverse the polygon twice, then twice in clockwise direction, and report that it is self-approaching.

Case 1. If only the first intersection type occurs, then the algorithm will traverse the boundary of P until the first violation of the half-strip property, correctly report that the polygon is not self-approaching, and terminate.

Case 2. Suppose that the second intersection type occurs. Consider the first segment $\overline{p_{i-1}p_i}$, such that both h_i and $\overline{p_{i-1}p_i}$ intersect ρ_r. Let $\overline{p_{i-1}p_i}$ intersect the normal to some preceding segment $\overline{p_{j-1}p_j}$ at the point p_j. As the ray h_i intersects ρ_r before $\overline{p_{i-1}p_i}$ does, it also intersects the polygon boundary between the points p_j and p_{i-1}. And, therefore, the ray h_{i-1} perpendicular to $\overline{p_{i-1}p_i}$ at the point p_{i-1} also intersects the polygon boundary between the points p_j and p_{i-1}. Then, consider the behavior of the algorithm during the backwards traversal. Let p_ℓ for $j \leq \ell < i-1$ be the first point on the left side of the ray h_{i-1}. Then the segment $\overline{p_{\ell+1}, p_\ell}$ intersects h_{i-1}, and either the segment $\overline{p_{\ell+1}, p_\ell}$ intersects the left chain ρ_l or there was another segment before $\overline{p_{\ell+1}, p_\ell}$ that intersected ρ_l. Note, that because the intersection of $\overline{p_{i-1}p_i}$ and ρ_r was the first violation of the half-strip property in counter-clockwise order, the intersection of $\overline{p_{\ell+1}, p_\ell}$ and ρ_l cannot be of the second type, otherwise p_{i-1} would already lie on the right side of a normal to $\overline{p_{\ell+1}, p_\ell}$ at the point $\overline{p_{\ell+1}}$. Therefore, this intersection can only be of type one, and the algorithm will recognize it during the backwards traversal.

Case 3. Suppose that the third intersection type occurs. Then, there will be a segment $\overline{p_{j+1}, p_j}$ (where $j \geq i$), for which the first or the second intersection type occurs when traversing the polygon in the opposite direction, and thus either case 1 or case 2 applies.

Thus, we only need to explicitly check for the first intersection type. The running time of the algorithm is $O(n)$. At every iteration, the number of segments removed from the list ρ_r is equal to half the number of tests for intersections the algorithm makes, and the number of segments added back is at most 2. Therefore, the total number of segments that can be removed from ρ_r over one traversal of the boundary is not more than $2n$. Similarly, the total number of segments that can be removed from ρ_l over one traversal of the boundary is not more than $2n$. Therefore, the algorithm performs $O(n)$ intersection tests. ◀

Acknowledgements. This work was begun at the CMO-BIRS Workshop on Searching and Routing in Discrete and Continuous Domains, October 11–16, 2015.

References

1 O. Aichholzer, F. Aurenhammer, C. Icking, R. Klein, E. Langetepe, and G. Rote. Generalized self-approaching curves. *Discrete Applied Mathematics*, 109(1-2):3–24, 2001. doi:10.1016/S0166-218X(00)00233-X.

2 S. Alamdari, T. M. Chan, E. Grant, A. Lubiw, and V. Pathak. Self-approaching Graphs. In *20th International Symposium on Graph Drawing (GD)*, pages 260–271, 2012. doi:10.1007/978-3-642-36763-2_23.

3 E. M. Arkin, R. Connelly, and J. S. B. Mitchell. On monotone paths among obstacles with applications to planning assemblies. In *5th Annual Symposium on Computational Geometry (SCG)*, pages 334–343. ACM Press, 1989. doi:10.1145/73833.73870.

4 M. A. Bender and M. Farach-Colton. The LCA Problem Revisited. In *Latin American Symposium on Theoretical Informatics*, pages 88–94, 2000. doi:10.1007/10719839_9.

5 M. Biro, J. Iwerks, I. Kostitsyna, and J. S. B. Mitchell. Beacon-Based Algorithms for Geometric Routing. In *13th Algorithms and Data Structures Symposium (WADS)*, pages 158–169. Springer, 2013. doi:10.1007/978-3-642-40104-6_14.

6 P. Bose, I. Kostitsyna, and S. Langerman. Self-approaching paths in simple polygons. Preprint, http://arxiv.org/abs/1703.06107, 2017.

7 B. Chazelle, H. Edelsbrunner, M. Grigni, L. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12(1):54–68, 1994. doi:10.1007/BF01377183.

8 H. Dehkordi, F. Frati, and J. Gudmundsson. Increasing-Chord Graphs On Point Sets. In *22nd International Symposium on Graph Drawing*, pages 464–475. Springer, 2014. doi:10.1007/978-3-662-45803-7_39.

9 L. E. Dubins. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *American Journal of Mathematics*, 79(3):497–516, 1957. doi:10.2307/2372560.

10 J. Gao and L. Guibas. Geometric algorithms for sensor networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1958):27–51, 2012. doi:10.1098/rsta.2011.0215.

11 L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2(1-4):209–233, 1987. doi:10.1007/BF01840360.

12 C. Icking and R. Klein. Searching for the kernel of a polygon – a competitive strategy. In *11th Annual Symposium on Computational Geometry (SCG)*, pages 258–266. ACM Press, 1995. doi:10.1145/220279.220307.

13 C. Icking, R. Klein, and E. Langetepe. Self-approaching curves. *Mathematical Proceedings of the Cambridge Philosophical Society*, 125(3):441–453, 1999. doi:10.1017/S0305004198003016.

14 M. Laczkovich. The removal of π from some undecidable problems involving elementary functions. *Proceedings of the American Mathematical Society*, 131(07):2235–2241, 2003. doi:10.1090/S0002-9939-02-06753-9.

15 J. S. B. Mitchell, C. Piatko, and E. M. Arkin. Computing a shortest k-link path in a polygon. In *33rd Annual Symposium on Foundations of Computer Science*, pages 573–582. IEEE, 1992. doi:10.1109/SFCS.1992.267794.

16 M. Nöllenburg, R. Prutkin, and I. Rutter. On self-approaching and increasing-chord drawings of 3-connected planar graphs. *Journal of Computational Geometry*, 7(1):47–69, 2016. doi:10.20382/jocg.v7i1a3.

17 G. Rote. Curves with increasing chords. *Mathematical Proceedings of the Cambridge Philosophical Society*, 115(01):1, 1994. doi:10.1017/S0305004100071875.