Self-Approaching Paths and Paths with Increasing Chords in Polygonal Domains

Hagedoorn, Mart

Award date:
2021

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science
Algorithms, Geometry, and Applications

Self-Approaching Paths and Paths with Increasing Chords in Polygonal Domains

Master's Thesis

Mart Hagedoorn

Committee:
Dr. Irina Kostitsyna
Prof. Dr. Mark de Berg
Dr. Ir. Tom Verhoeff

Supervisor:
Dr. Irina Kostitsyna

Eindhoven, November 2021

Abstract

In this thesis we study two types of constrained paths in different polygonal settings, namely *self-approaching* paths and paths with *increasing chords*. A directed path is self-approaching if the distance between a point traveling along the path and a point that has not been traversed yet is not decreasing; more specifically for any points a, b, and c that lie in that order on the path, $|ac| \geq |bc|$. Furthermore, a path has increasing chords if the path is self-approaching in both directions, that is for any points a, b, c, and d that lie in that order on the path, $|ad| \geq |bc|$.

Existing literature on this subject describes an algorithm for finding a shortest self-approaching path inside a simple polygon. In this thesis, we show that finding the shortest path with increasing chords is similar to finding the shortest self-approaching path. Furthermore, we propose an algorithm that can find the shortest self-approaching path in a polygon with holes, albeit inefficiently. Lastly, we show that if any two points in the polygon can be connected by a self-approaching path, the holes must have all centers of curvatures internal to the holes. If any two points can be connected by a path with increasing chords, the holes must be of constant width.

Contents

Contents **iii**

1 Introduction **1**
1.1 Related work 2
1.2 Contributions 3

2 Preliminaries **4**

3 Paths with Increasing Chords in Simple Polygons **10**
3.1 An algorithm for paths with increasing chords 16

4 Self-Approaching Paths in Polygonal Domains with Holes **19**
4.1 Exploration for trivial self-approaching algorithms 19
4.1.1 Shortest geodesic 19
4.1.2 Shortest geodesic with self-approaching path 20
4.2 Self-approaching algorithm for holes 22

5 Characterization of Strongly Connected Polygons **25**
5.1 Self-approaching polygons 25
5.2 Polygons with increasing chords 27

6 Discussion **32**
6.1 Future work 32

Bibliography **34**

Appendix **35**

A Finding the ratio between two distinct shortest self-approaching paths **36**

Chapter 1

Introduction

In the area of theoretical computer science, finding an obstacle-avoiding path between two points in a geometric domain has long been a prominent question. Whilst obstacle avoidance can be a constraint imposed on path finding problems, sometimes the path has to conform to more constraints to meet specific requirements. For example, with beacon and greedy routing *radially monotone* paths appear. Radially monotone paths are directed curves such that the distance between a point traversing the path and the destination is never increasing [4, 3, 8].

A more restricted version of radially monotone paths are *self-approaching* paths: a directed path such that the distance between a point traversing the path and any point on the remainder of the path is decreasing. More formally, a path π is self-approaching if and only if for any points a, b, and c that appear in that order on π, the Euclidean distance between a and c is greater or equal than the Euclidean distance between b and c [13]. Trivially, self-approaching paths are radially monotone (let c be the destination of the path). Moreover, every subpath of a self-approaching path is radially monotone, which can be beneficial for situations where the destination is unknown. Furthermore, the length of a self-approaching path is bounded in comparison with the Euclidean distance between the start and destination of the path, which is not the case for radially monotone paths.

A strengthening of self-approaching paths are paths with *increasing chords*; a path has increasing chords if and only if the path is self-approaching in both directions. Again more formally, a path σ has increasing chords if and only if for any points a, b, c, and d that appear in that order on σ, the Euclidean distance between a and d is greater or equal than the Euclidean distance between b and c [14]. Therefore, a path with increasing chords can be used when an agent has to greedily route to an endpoint and thereafter be greedily routed back to the start point. Moreover, the length of a path with increasing chords is bounded even more than with self-approaching paths when compared with the Euclidean distance between the start and destination of the path.

Having an improved understanding of self-approaching paths and paths with increasing chords might help in finding better solutions to questions in sensor networks, graph drawings, and constrained path problems in general. There are still plenty of open questions concerning self-approaching paths and paths with increasing chords. We aim to answer four of these open questions.

- Given two points s and t in a simple polygon P, find the shortest path with increasing chords inside P if it exists.
- Given two points s and t in polygon P with holes, find a shortest self-approaching path inside P if it exists.
- Given a polygon P with holes, test if there exists a self-approaching path between any two points in P.
- Given a polygon P with holes, test if there exists a path with increasing chords between any two points in P.

1.1 Related work

From the more complicated paths described above we first revisit the arguably simplest type of constrained paths which is the *geodesic* inside a polygon. A geodesic is the shortest path connecting two points that lie inside a polygon. Finding the geodesic is a problem that can be solved in $\mathcal{O}(n)$, given a triangulation of a simple polygon P with n vertices [10]. The triangulation of a simple polygon can be found with a running time of $\mathcal{O}(n)$ [7]. Furthermore, finding a geodesic in a polygon P with holes with n vertices costs $\mathcal{O}(n \log n)$, making this problem slightly harder to solve. Trivially, as long as two points $s, t \in P$ belong to the same connected component, a geodesic connecting s and t exists.

A more intricate type of constrained paths are the aforementioned self-approaching paths as were introduced by Icking et al. [13]. Bose et al. [5] provided an algorithm that can find a shortest self-approaching path in a simple polygon P connecting points s and t. However, the problem of finding a self-approaching path in a polygon P with holes remains open. Furthermore, Bose et al. proposed an efficient algorithm that tests whether in a simple polygon between any two points a self-approaching path can be constructed. Testing whether in a polygon with holes between any two points a self-approaching path can be constructed remained open.

Paths with increasing chords are closely related to self-approaching paths. However, at the time of writing no algorithm existed that finds the shortest path with increasing chords in a simple polygon. Therefore, also no algorithm exists that finds the shortest paths with increasing chords in a polygon with holes. If a self-approaching path between any two points in a simple polygon P exists, then between any two points a path with increasing chords also must exist [5]. Hence, to test if it is possible to construct a path with increasing chords between any two points in a simple polygon P, testing if between any two points a self-approaching path can be constructed is sufficient.

Some research has focused on finding the *stretching factor*, i.e. the maximum ratio between the length of a constrained path γ and the Euclidean distance between the endpoints of γ. Since the only constraint on a geodesic is that it must fully lie inside a polygon P, the stretching factor does not exist. For self-approaching paths a stretching factor of approximately 5.3331 exists [13]. Later, Aichholzer et al. [1] extended on this work by providing a function $c(\varphi)$ that gives the stretching factor for *φ-self-approaching path*. A φ-self-approaching path is a path such that the entire remainder after any point p always falls in a wedge with angle φ and apex p. For paths with increasing chords, the stretching factor was proven to be $2\pi/3$ [21]. Yet, no function is found that gives the stretching factor for paths that are φ-self-approaching in both directions.

Furthermore, some research has been done that is more application focused. A field closely related to self-approaching paths and paths with increasing chords is greedy routing in geometric networks, which has received a considerable amount of attention [2, 9, 22, 15, 18]. The goal in such problems is to send a packet between any two vertices s and t in a network, such that the path can be greedily constructed. Paths with increasing chords were introduced to put an upper bound on the ratio of the aforementioned $2\pi/3$ between the length of a path and the Euclidean distance between the start and end points. Another field closely related to self-approaching paths is the field of beacon based routing and kernel searching [4]. For example, self-approaching paths can be used for analysing the strategy of mobile robots searching for a kernel as is shown by Icking et al. [12]. Finally, Biro et al. [3] showed that finding the minimal set of beacons that cover a simple polygon is NP-hard. This problem can be directly translated into finding the minimum number of target points, such that any point in a simple polygon can be connected by a self-approaching path to one of the target points.

Paths with increasing chords can also be studied in graph settings. For example, when given a point set with n points, a drawing with increasing chords with $\mathcal{O}(n)$ Steiner points can be constructed that spans all points [19]. A drawing of a graph has increasing chords if any pair of vertices can be connected by a path along the edges of the graph with increasing chords. Furthermore, Nöllenburg et al. [17] show how to construct planar increasing-chord drawings for planar 3-trees.

1.2 Contributions

In Chapter 3, we prove that given a polygon P and points $s, t \in P$, a shortest s-t path with increasing chords can be calculated using methods proposed by Bose et al. [5]. In Chapter 4, we discuss why finding a shortest self-approaching path in a polygon with holes is not trivial. Furthermore, we propose an algorithm that can find the shortest self-approaching path in a polygon with holes, albeit inefficiently. In Chapter 5, we show that if any two points in a polygon with holes can be connected by a self-approaching path, the holes must have all centers of curvatures internal to the holes. If any two points can be connected by a path with increasing chords, the holes must be of constant width.

Chapter 2

Preliminaries

In this section we will provide definitions and properties that will prove to be useful later. The concepts described in this section will remain central for the remainder of the thesis. All definitions and concepts are for the two dimensional setting.

Paths

An s-t path ζ is a directed curve that starts in point s and ends in point t, such that ζ lies entirely inside of polygon P, i.e. $\zeta \subseteq P$. For any two points p and q on path ζ, we use $p \leq_\zeta q$ to denote that either p comes before q on ζ or $p = q$. We use $\zeta(p, q)$ to denote the subpath of ζ between points p and q. Furthermore, $|pq|$ gives us the Euclidean distance between points p and q.

Paths in an Euclidean space can make smooth turns and sharp turns. In order to differentiate between these types of turns, we use the standard concept of a *bend point* (Figure 2.1).

Definition 1. A *bend point* b of piecewise smooth curve ζ is a point where the first derivative of ζ is discontinuous.

We define a *normal* h_p to ζ at point p to be the line through a given point $p \in \zeta$ such that this line is perpendicular to the tangent of ζ in p. If p is a bend point, we use the normal definition as proposed by Icking et al. [13]. This definition states that the normal to p is the set of lines that are included in the double wedge of the perpendiculars to the tangents of the smooth pieces meeting at p (Figure 2.1). A normal h_p to path ζ defines two half-planes, if p is a bend point every line in the wedge defines two half-planes which can all be considered separately. Let the *negative half-plane* h_p^- be the closed half-plane that is defined by h_p and congruent with the opposite direction of ζ in p. Furthermore, let the *positive half-plane* h_p^+ be the closed half-plane that is defined by h_p and congruent with the direction of ζ in p (Figure 2.2).

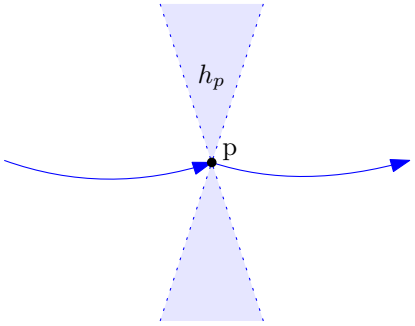


Figure 2.1: The normal h_p of p, where p is a bend point

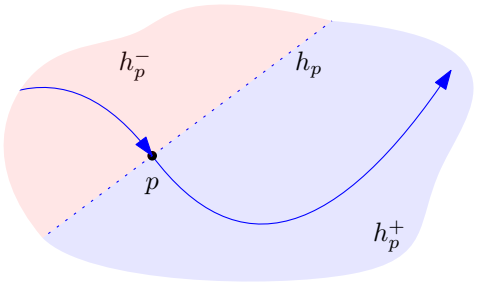


Figure 2.2: Example figure showing the normal h_p at point p and the corresponding half planes h_p^- and h_p^+

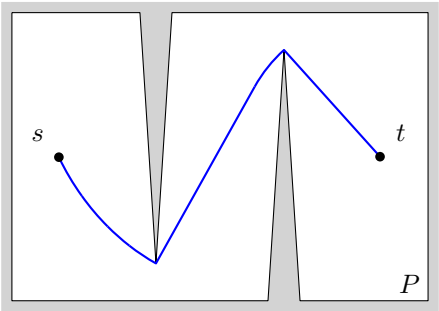


Figure 2.3: A self-approaching s-t path inside polygon P

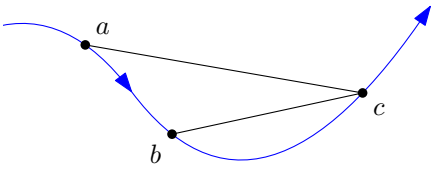


Figure 2.4: Three points on a directed path such that $|ac| \geq |bc|$.

Self-approaching paths An often used intuitive definition for self-approaching paths is a path π that, whilst traversing along π, one gets closer to each point of the curve that has not yet been passed. Figure 2.3 shows an example of a self-approaching path in polygon P. The formal definition for self-approaching paths was given by Icking et al. [13].

Definition 2. An s-t path π is *self-approaching* if and only if for any three sequential points $a, b, c \in \pi$, such that $a \leq_\pi b \leq_\pi c$, $|ac| \geq |bc|$, i.e. the Euclidean distance between a and c is greater or equal than the Euclidean distance between b and c (Figure 2.4).

Icking et al. also proved the normal property for self-approaching paths. For any point p on a self-approaching path π, the remainder of the path after p does not intersect the normal to π at p.

Lemma 1 (Normal property [13])**.** *An s-t path π is self-approaching if and only if any normal h_p to π at any point $p \in \pi$ does not intersect with $\pi(p, t)$.*

The normal property can be reformulated to a notation using half planes. We can reformulate Lemma 1 as follows:

Lemma 2 (Half-plane property)**.** *An s-t path π is self-approaching if and only for any positive half-plane h_p^+ at point $p \in \pi$, $\pi(p, t)$ lies completely in h_p^+.*

A *perpendicular bisector* of a line segment is the line which intersects that line segment through its midpoint perpendicularly. A self-approaching s-t path π has the property that the perpendicular bisector of the line segment between two sequential points p and q does not intersect the remaining subpath $\pi(q, t)$.

Lemma 3 ([5])**.** *For any two points $p <_\pi q$ on a self-approaching s-t path π in $\mathbb{R}^2$, the perpendicular bisector of the straight-line segment pq does not intersect the subpath $\pi(q, t)$.*

Paths with increasing chords A strengthening on the restriction of self-approaching paths is the restriction upon paths with increasing chords. Figure 2.5 shows an example of an s-t path with increasing chords in polygon P. The formal definition of a path with increasing chords is:

Definition 3. An s-t path π has *increasing chords* if and only if for any four sequential points $a, b, c, d \in \pi$, such that $a \leq_\pi b \leq_\pi c \leq_\pi d$, $|ad| \geq |bc|$, i.e. the Euclidean distance between a and d is greater or equal than the Euclidean distance between b and c (Figure 2.6).

It is well known that a path has increasing chords if and only if that path is self-approaching in both directions, for completeness we provide the proof here.

Lemma 4. *An s-t path σ has increasing chords if and only if σ is a self-approaching s-t and t-s path.*

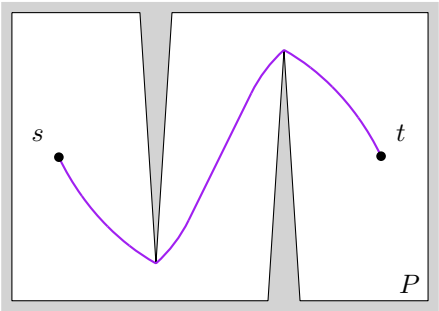


Figure 2.5: An s-t path with increasing chords inside polygon P

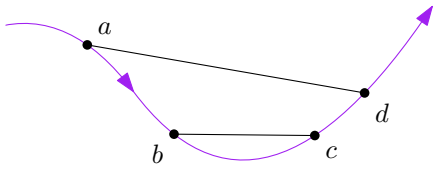


Figure 2.6: Four points on a directed path such that $|ad| \geq |bc|$.

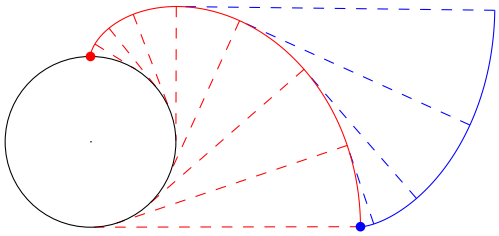


Figure 2.7: The red involute is the evolute of the blue involute and the black circle is the evolute of the red involute. The black circle is an involute of order 0, the red involute is of order 1 and the blue involute is of order 2.

Proof. If σ has increasing chords, then for any four points $a, b, c, d \in \sigma$, that appear in that order on σ, $|ad| \geq |bc|$ must hold. Therefore, when $c = d$ the inequality must still hold and we get the property $|ac| \geq |bc|$. Hence, σ is a self-approaching s-t path. Furthermore, when $a = b$, we see that $|bd| \geq |bc|$ must hold, and therefore σ must be also a self-approaching t-s path.

If σ is a self-approaching s-t and t-s path, then for points $a, b, c, d \in \sigma$ and $a \leq_\sigma b \leq_\sigma c \leq_\sigma d$, $|ac| \geq |bc|$ and $|ad| \geq |ac|$ must hold. Therefore, $|ad| \geq |bc|$ holds and σ has increasing chords. □

Often the normal and half-plane properties are used for paths with increasing chords. These properties follow almost directly from the normal property of self-approaching paths and Lemma 4. Again, for completeness we provide the proof for the normal property of paths with increasing chords.

Lemma 5 (Normal property)**.** *An s-t path σ has increasing chords if and only if any normal to σ at any point $p \in \sigma$ does not intersect with $\sigma(s, p)$ and $\sigma(p, t)$.*

Proof. This normal property follows almost directly from Lemma 4 and the normal property of a self-approaching path. At any point $p \in \sigma$, the normal properties of the self-approaching s-t and t-s paths state that $\sigma(p, t)$ and $\sigma(s, p)$ cannot cross the normal to σ through p. □

This normal property can be rewritten into the half-plane property for paths with increasing chords.

Corollary 6 (Half-plane property)**.** *An s-t path σ has increasing chords if and only if, for any line h normal to σ at any point $p \in \sigma$, the subpath $\sigma(s, p)$ lies completely in the negative half-plane h_p^- and the subpath $\sigma(p, t)$ lies completely in the positive half-plane h_p^+.*

Involutes

This thesis will also often use the notion of an involute. Intuitively, an involute is obtained by following a point on a piece of string that is either being wrapped or unwrapped around a given

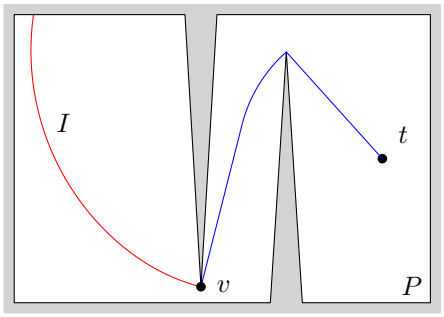


Figure 2.8: For any point s on involute I (red) of the convex hull of shortest self-approaching v-t path π (blue), $I(s, v) \oplus \pi$ is the shortest self-approaching s-t path.

curve (Figure 2.7). The formal definition is as follows:

Definition 4 ([16])**.** An *involute* I of a smooth curve ζ is a curve such that point $p \in I$ lies on the tangent line to ζ at point $q \in \zeta$ and a vector in the direction of the first derivative of p is perpendicular to that of q. An *evolute* of a curve ζ is the locus of all centers of curvatures of ζ. Moreover, ζ is the evolute of I.

Since every normal of an involute I of a curve ζ touches ζ these types of curves are excellent for finding shortest self-approaching paths. For example, let π be a shortest self-approaching v-t path and I an involute of the convex hull of π that goes through v. For any point $s \in I$ the concatenation $I(s, v) \oplus \pi$ is the shortest self-approaching s-t path [5] (Figure 2.8).

Furthermore, we will use the notion of involutes of th kth order as was introduced by Bose et al. [5]. We say that an involute is of *order* k if it is an involute of a smooth involute of order $k-1$ (Figure 2.7). Furthermore, we say circular arcs are involutes of order 0. Thus, an involute of a circular arc is of order 1 and so forth. An involute I_k of order k is defined as:

$$I_k(\theta) = \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} (-1)^i a_{2i}(\theta) \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} - \sum_{i=1}^{\lceil \frac{k}{2} \rceil} (-1)^{i-1} a_{2i-1}(\theta) \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix}, \tag{2.1}$$

where

$$a_i(\theta) = r_0 \frac{\theta^i}{i!} + c_1 \frac{\theta^{i-1}}{(i-1)!} + c_2 \frac{\theta^{i-2}}{(i-2)!} + \cdots + c_i \tag{2.2}$$

for some constants $c_1, c_2, \cdots, c_i$. Given a point $p_i(r_i, \varphi_i)$ where r_i and φ_i are the polar coordinates for p_i, the center of this coordinate center is the center of the involute of order 0, i.e. the circular arc. For every order of involute i we can calculate constant c_i with the following formulas:

$$\begin{aligned} r_i \cos(\theta_i - \varphi_i) &= a_0(\theta_i) - a_2(\theta_i) + \cdots \\ r_i \sin(\theta_i - \varphi_i) &= a_1(\theta_i) - a_3(\theta_i) + \cdots \end{aligned} \tag{2.3}$$

When $i \geq 2$, Equation 2.3 can no longer be solved analytically as these equations are transcendental. Therefore, to solve such an equation, numerical methods can be applied to find an approximation of a higher order involute. An example of such an approximation can be seen in Appendix A.

Dead regions

Next, we introduce the notion of *dead regions* [5]. These regions can be constructed with the aforementioned involutes. Furthermore, dead regions will help us in finding the shortest self-approaching path in a simple polygon.

Definition 5. Let dead region $\mathcal{D}_t$ for point t be the set of points, such that for any point $s \in \mathcal{D}_t$ no self-approaching s-t path exists (see Figure 2.9).

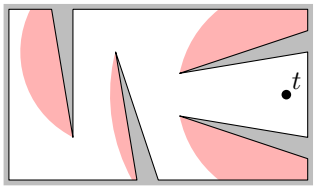


Figure 2.9: Dead region $\mathcal{D}_t$ (red) for some point t in a simple polygon.

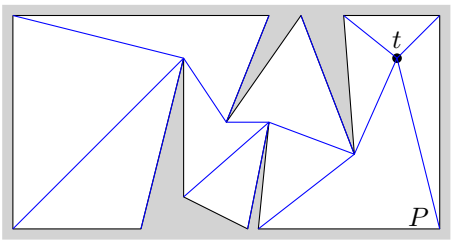


Figure 2.10: Example of the shortest path tree SPT_t (blue) containing the union all shortest paths between vertices of simple polygon P and t.

Bose et al. describe how to build dead region $\mathcal{D}_t$ given a point t that lies inside simple polygon P. First, we define SPT_t to be the shortest-path tree of t in P, i.e. the union of all shortest paths between t and every vertex of P (Figure 2.10). Every interior node p_i of SPT_t is reflex towards the interior of P. Let π_{p_it} be the shortest self-approaching path from p_i to t, if this path exists, and $CH(\pi_{p_it})$ the convex hull of π_{p_it}. We can start constructing the involute I of $CH(\pi_{p_it})$ starts in p_i and rotates into the same direction as the rotation of the shortest path through p_i. We construct I until it intersects some edge of P. Let $S(p_i)$ be the area on the concave side of the involute that is cuts off of the polygon P. We call $S(p_i)$ the *shadow* of p_i. Lemma 7 shows that indeed for any point v in shadow $S(p_i)$ of a point p_i, there does not exist a self-approaching v-t path.

Lemma 7 ([5]). *$S(p_i) \subseteq \mathcal{D}_t$.*

Moreover, Lemma 8 tells us that we can find whole dead region $\mathcal{D}_t$ by only calculating the shadows of interior nodes of the shortest-path tree SPT_t of t.

Lemma 8 ([5]). *$\mathcal{D}_t = \bigcup S(p_i)$ over all interior nodes p_i of the shortest-path tree SPT_t of t.*

Shortest self-approaching path

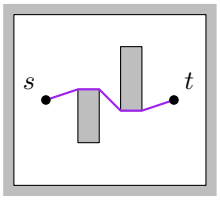
Most of the aforementioned lemmas and definitions were used to obtain Theorem 9 that gives us a description on how to find the shortest self-approaching s-t path.

Theorem 9 ([5]). *A shortest self-approaching s-t path in a simple polygon P is the geodesic path between s and t inside the region $P \setminus \mathcal{D}_t$.*

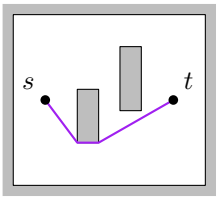
Furthermore, Bose et al. [5] showed that a shortest self-approaching path in a simple polygon consists of straight line-segments, circular arcs, and involutes of some order k. A *segment* of a path that is defined by one function, i.e. any of the three aforementioned parts of a path are separate segments. The number of segment in a shortest self-approaching s-t in a simple polygon P can be bounded to the number of vertices of P.

Lemma 10 ([5]). *A shortest self-approaching s-t path in a simple polygon P with n vertices consists of $\Theta(n^2)$ segments.*

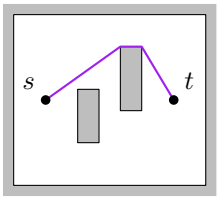
Using Theorem 9 and 10 Bose et al. [5] proposed an algorithm that can find the shortest self-approaching s-t path in a simple polygon P in $\mathcal{O}(n \cdot f(k) + n \cdot g(k))$, where n is the number of



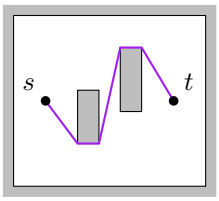
(a) The simple 1-path.



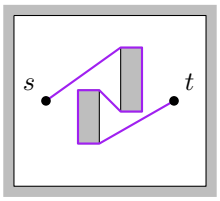
(b) The simple 2-path.



(c) The simple 3-path.



(d) The simple 4-path.



(e) The simple 5-path.

Figure 2.11: Every shortest simple k-path in a polygon with holes.

vertices of P and k the order of the highest involute that appears in the solution. Since involutes of order higher than 2 cannot be solved analytically, $f(k)$ and $g(k)$ represent the computing time for solving and evaluating Equation 2.3, respectively.

Holes

Finally, we introduce the notion of holes. A *hole* H is a connected open set of points that lies interior to P but does not belong to polygon P. Therefore, the boundaries of H and P cannot intersect, i.e. there always exists a path around H inside P.

We say two s-t paths are *homotopically equivalent* if one path can be continuously deformed into the other path without intersecting the boundary of polygon P. When a polygon has holes multiple paths can exist between two end points. To identify these paths we use the definition for a simple k-path provided by Hershberger et al. [11]. The simple 1-path is the shortest path between s and t. The simple 2-path is the shortest path which does not cross itself and is not homotopically equivalent to the 1-path. Thereafter, the simple k-path is defined recursively (Figure 2.11). To better match the terminology used in this thesis, we will use the term k-geodesic instead of a simple k-path. Finally, we say that a curve ζ is shortest in its *homotopy group* if there does not exist another curve that is shorter and homotopically equivalent to ζ.

Chapter 3

Paths with Increasing Chords in Simple Polygons

In this chapter we will propose an algorithm that can find a path with increasing chord if it exists. Recall that a path with increasing chords is a path such that the Euclidean distance between two points is greater or equal than the Euclidean distance between two other points that lie on the subpath between the two former points. Paths that have increasing chords are closely related to self-approaching paths. As a consequence, we will often use the proprieties from self-approaching paths to prove properties of paths with increasing chords. Using the normal and half-plane property we can start drawing conclusions about shortest paths with increasing chords.

We will first show that a shortest s-t path with increasing chords is unique in a simple polygon P, therefore we only need to search for one shortest s-t path with increasing chords. The following two proofs are extensions from the analogous propositions about self-approaching paths given by Bose et al. [5].

Lemma 11. *A geodesic path γ between two distinct paths with increasing chords σ_1 and σ_2 also has increasing chords (Figure 3.1).*

Proof. First of all, we will use the fact that the geodesic must lie in both convex hulls of both paths, i.e. $\gamma \subseteq CH(\sigma_1)$ and $\gamma \subseteq CH(\sigma_2)$. Furthermore, any point $p \in \gamma$ either lies on one of the paths σ_1 or σ_2 or on a straight line that is bitangent to σ_1, σ_2 or both σ_1 and σ_2.

First, we consider the case that p is not a bend point but lies on a smooth section of σ_1 or σ_2. Let, w.l.o.g., $p \in \sigma_1$. The positive half-plane h_p^+ of the normal to σ_1 at p contains subpath $\sigma_1(p,t)$ and the negative half-plane h_p^- of the normal at p contains subpath $\sigma_1(s,p)$. Therefore, h_p^+ also contains the convex hull of $\sigma_1(p,t)$ and h_p^- contains the convex hull of $\sigma_1(s,p)$. Hence, h_p^+ contains the subpath $\gamma(p,t)$ and h_p^- contains the subpath $\gamma(s,p)$ of the geodesic.

Now let p be a bend point lying on σ_1. Subpath $\sigma_1(p,t)$ must be fully contained in the two positive half-planes, which are defined by the normals of the smooth pieces of σ_1 meeting at p. Analogously $\sigma_1(s,p)$ is fully contained in the two negative half-planes. The two normals of the geodesic path at this point must also lie in between the two normals to the boundary path. Thus,

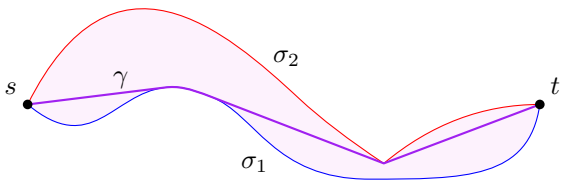


Figure 3.1: Geodesic γ (purple) that lies between paths σ_1 (blue) and σ_2 (red) with increasing chords.

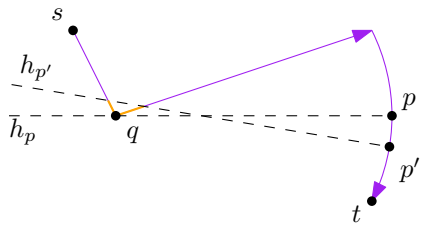


Figure 3.2: Geodesic σ (purple) where the normal property does not hold in p', normal h_p touches point q that lies on subpath s (orange) of σ.

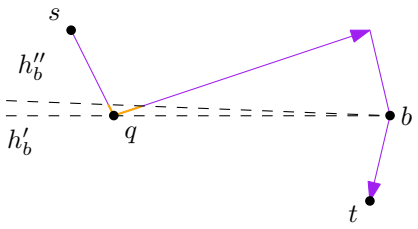


Figure 3.3: Geodesic σ (purple) where the normal property does not hold in p', line h'_b touches point q that lies on subpath s (orange) of σ.

the intersection of the two positive half-planes of the normals to the geodesic contains the convex hull of the subpath $\sigma_1(p,t)$, and, therefore, the rest of the geodesic path $\gamma(p,t)$. Analogously, the intersection of the two negative half-planes of the normals to the geodesic contains the convex hull of the subpath $\sigma_1(s,p)$, and, therefore, the geodesic subpath $\gamma(s,p)$.

Finally, let p lie on a segment of γ that is bitangent to σ_1, σ_2 or both σ_1 and σ_2. Let us consider one of the end-points p^* of this bitangent. Assume, that $p^* \in \gamma(p,t)$. The normal to γ at p is parallel to the normal to γ at p^*. By one of the cases considered above, the positive half-plane at p^* will contain $\gamma(p^*,t)$, and, therefore, the positive half-plane of the normal to γ at p will also contain the subpath $\gamma(p,t)$. If $p^* \in \gamma(s,p)$, we can follow analogues steps to show that $\gamma(s,p)$ lies in the negative half-plane of p.

Thus, γ has increasing chords. □

Using the property from Lemma 11 we can now prove the following theorem.

Theorem 12. *A shortest s-t path with increasing chords in a simple polygon is unique.*

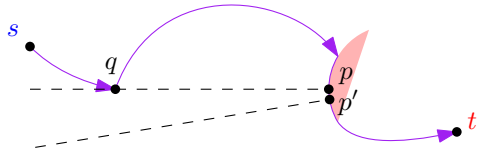
Proof. Assume there exist two distinct shortest s-t paths with increasing chords σ_1 and σ_2 in a polygon P. Then, by Lemma 11, there exists a shorter path that is a geodesic path enclosed between σ_1 and σ_2. □

Recall that $\mathcal{D}_t$ for point t is the set of points, such that for any point $s \in \mathcal{D}_t$ no self-approaching s-t path exists in polygon P. In the next theorem we will show that if we subtract the union of $\mathcal{D}_t$ and $\mathcal{D}_s$ from simple polygon P, the geodesic shortest s-t path in this space is the shortest path with increasing chords, i.e. the region for which from every point a self-approaching path to s and t exists. The theorem is proven by contradiction. In more detail, we show that if there is a point where the normal property is violated, some self-approaching path cannot exist.

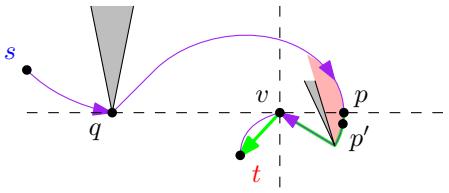
Theorem 13. *Geodesic σ between points s and t inside the region $P \setminus (\mathcal{D}_t \cup \mathcal{D}_s)$ is the shortest s-t path with increasing chords in simple polygon P.*

Proof. Assume that σ does not have increasing chords. Let point $p \in \sigma$ be such that p is the last point along σ for which the normal property holds if p exists. If p exists, there also exists point $p' \in \sigma(p,t)$ such that p' lies in the ϵ-neighborhood of p for arbitrary small ϵ and the normal property does not hold in p'. Since in p' the normal property does not hold, there is some subpath ρ of $\sigma(s,p')$ or $\sigma(p',t)$ that either lies in the positive or negative half-plane defined by the normal through h'_p, respectively. Furthermore, the normal h_p touches σ at some point $q \in \rho$ (Figure 3.2). On the contrary if p does not exist, there exists a bend point b such that there are two lines $h'_b, h''_b \in h_b$ among the set of lines in the normal h_b, such that h'_b touches σ at some point $q \in \sigma$ and h''_b intersect σ (Figure 3.3). All cases where p does not exist are simply analogous to the cases where p exists. Therefore, we will only cover the cases where p exists.

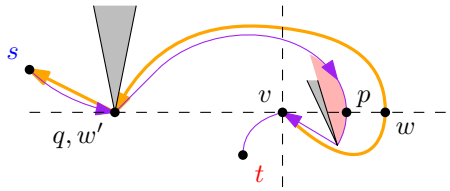
We assume, w.l.o.g, that line segment $\overline{pq}$ is horizontal, p lies to the right of q, and $\sigma(s,p)$ lies above h_p. Let e be the segment of σ p lies on and f be the segment of σ q lies on. We must



(a) The point of curvature of σ at point p lies to the right of p.



(b) The point of curvature v of σ at point p lies to the left of p. The green path is the shortest self-approaching p-t path.



(c) The point of curvature v of σ at point p lies to the left of p. The orange path is the shortest self-approaching v-s path, that intersects with h_p in w and w'.

Figure 3.4: Geodesic σ (purple) where point $q \in \sigma(s,p)$, p lies on a boundary of $\mathcal{D}_t$ (red), and q is a vertex of P.

consider the cases where $q \in \sigma(s,p)$ and $q \in \sigma(p,t)$. Furthermore, the segments e and f can be straight line segments, or boundaries of a dead region of $\mathcal{D}_t$ or $\mathcal{D}_s$. If f is a straight line segment, since σ is a geodesic, q must be an end point of f and thus be a vertex of polygon P. We will go over every case separately and show that these cases are not possible.

Once we have shown that no point exists where the normal property is violated, σ indeed has increasing chords by Lemma 5. Furthermore, σ must be the shortest path with increasing chords. Any path that is shorter than σ must go through either of the dead regions $\mathcal{D}_t$ or $\mathcal{D}_s$. Therefore, any path shorter than σ cannot have increasing chords and σ must be the shortest path with increasing chords. We will now consider all different cases:

Case 1: $q \in \sigma(s,p)$, e is a boundary of $\mathcal{D}_t$, and q is a vertex of P (Figure 3.4)
The center of curvature of σ at p must lie in the direction of q. Otherwise the normal $h_{p'}$ to σ cannot intersect $\sigma(s,p)$, as is depicted in Figure 3.4a. Let π_{pt} be the shortest self-approaching path from p to t. Segment e is part of $\mathcal{D}_t$, therefore e is an involute of π_{pt} by construction. Hence, there must be a point $v \in \pi_{pt}$ touched by normal h_p (Figure 3.4b). Because π_{pt} goes through v, $h_v \perp h_p$, i.e. h_v is perpendicular to h_p (if v is a bend point, there must be a line which is perpendicular h_p in the set of normals h_v). Furthermore, $\pi_{pt}(v,t)$ lies completely in the positive half-plane h_v^+ by the half-plane property. We will now show that in the region between $\overline{pv}$ and $\pi_{pt}(p,v)$ there are vertices of P. The straight line segment $\overline{pv}$ concatenated to $\pi_{pt}(v,t)$ would be self-approaching. However, p' lies below h_p, thus polygon P must intersect with $\overline{vp}$. Therefore, the shortest self-approaching v-s path π_{vs} must first intersect or touch h_p to the right of v at point w and later to the left of v at point w' (Figure 3.4c). Thus, $|vw'| < |ww'|$ which contradicts the self-approaching property of π_{vs}. Hence, this case is not possible if the geodesic between s and t exists.

Case 2: $q \in \sigma(s,p)$, e is a boundary of $\mathcal{D}_t$, and f is a boundary of $\mathcal{D}_s$ (Figure 3.5)
This case is analogous to Case 1 (Figure 3.4).

Case 3: $q \in \sigma(s,p)$, e is a boundary of $\mathcal{D}_t$, and f is a boundary of $\mathcal{D}_t$ (Figure 3.6)
Since q is touched by h_p and q lies on a boundary of $\mathcal{D}_t$, normal h_q is perpendicular to h_p. Furthermore, consider shortest self-approaching q-t path π_{qt}. Because q lies on a boundary of $\mathcal{D}_t$

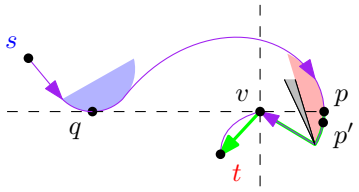


Figure 3.5: Point $q \in \sigma(s,p)$, p lies on a boundary of $\mathcal{D}_t$ (red), and q lies on a boundary of $\mathcal{D}_s$ (blue). The green path is the shortest self-approaching p-t path.

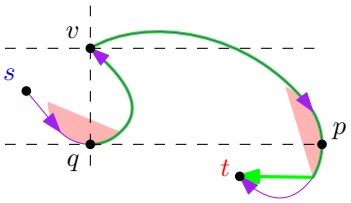


Figure 3.6: Point $q \in \sigma(s,p)$, and both p and q lie on a boundary of $\mathcal{D}_t$ (red). The green path is the shortest self-approaching q-t path.

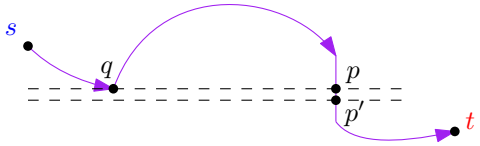


Figure 3.7: Point $q \in \sigma(s,p)$ and p lies on a straight line segment (orange).

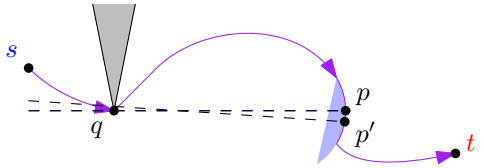


Figure 3.8: Point $q \in \sigma(s,p)$, p lies on a boundary of $\mathcal{D}_s$ (blue), and q is a vertex of P.

there is a point $v \in \pi_{qt}$ touched by h_q with $h_v \perp h_q$. Path $\pi_{qt}(p,t)$ or $\pi_{qt}(v,t)$ must lie both in h_v^+ and in h_p^+, however the intersection of these half-planes is empty; $h_v^+ \cap h_p^+ = \emptyset$. Hence, path π_{qt} cannot exist.

Case 4: $q \in \sigma(s,p)$ and e is a straight line segment (Figure 3.7)
Since the normals h_p and $h_{p'}$ are parallel it is impossible for $h_{p'}$ to be able to intersect with $\sigma(s,p)$.

Case 5: $q \in \sigma(s,p)$, e is a boundary of $\mathcal{D}_s$, and q is a vertex of P (Figure 3.8)
Normal h_p touches $\sigma(s,p)$ at point q. Point q does not lie in the positive half-plane $h_{p'}^+$ of the shortest self-approaching p'-s path $\pi_{p's}$. Since q is a vertex of P, $h_{p'}$ intersects with P. Therefore, s and p' must lie in different connected components in the intersection of P and the positive half-plane $h_{p'}^+$ of the path $\pi_{p's}$. Hence, no p'-s path can exist including $\pi_{p's}$.

Case 6: $q \in \sigma(s,p)$, e is a boundary of $\mathcal{D}_s$, and q is a boundary of $\mathcal{D}_s$ (Figure 3.9).
Normal h_p touches $\sigma(s,p)$ at point q. Point q does not lie in the positive half-plane $h_{p'}^+$ of the shortest self-approaching p'-s path $\pi_{p's}$. Since q lies on a boundary of $\mathcal{D}_s$, $h_{p'}$ intersects with this

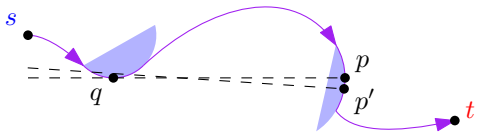


Figure 3.9: Point $q \in \sigma(s,p)$, and both p and q lie on a boundary of $\mathcal{D}_s$ (blue).

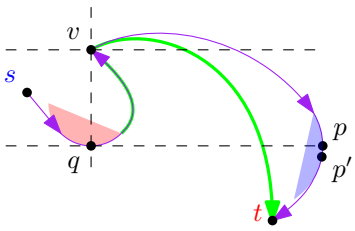


Figure 3.10: Point $q \in \sigma(s,p)$, p lies on a boundary of $\mathcal{D}_s$ (blue), and q lies on a boundary of $\mathcal{D}_t$ (red). The green path is the shortest self-approaching q-t path.

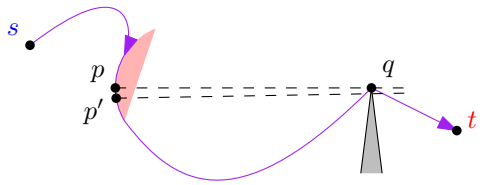


Figure 3.11: Point $q \in \sigma(p,t)$, p lies on a boundary of $\mathcal{D}_t$ (red), and q is a vertex of P.

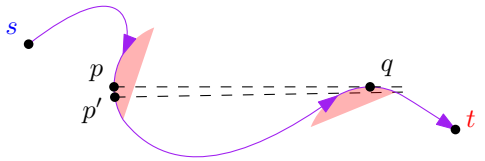


Figure 3.12: Point $q \in \sigma(p,t)$, and both p and q lie on a boundary of $\mathcal{D}_t$ (red).

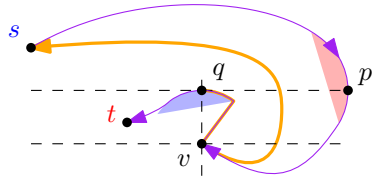


Figure 3.13: Point $q \in \sigma(p,t)$, p lies on a boundary of $\mathcal{D}_t$ (red), and q lies on a boundary of $\mathcal{D}_s$ (blue). The orange path is the shortest self-approaching q-s path.

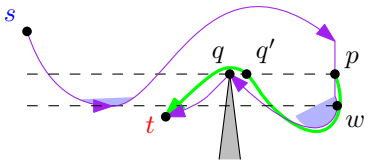


Figure 3.14: Point $q \in \sigma(p,t)$, both p and q lie on a straight line segment (orange), and point w lies on boundary d of $\mathcal{D}_s$ (blue) with $h_p \parallel h_w$. The green path is the shortest self-approaching p-t path.

boundary. Therefore, s and p' must lie in different connected components in the intersection of $P \setminus \mathcal{D}_s$ and the positive half-plane $h^+_{p'}$ of the path $\pi_{p's}$. Hence, $\pi_{p's}$ cannot exist.

Case 7: $q \in \sigma(s,p)$, e is a boundary of $\mathcal{D}_s$, and f is a boundary of $\mathcal{D}_t$ (Figure 3.10)
This case is analogous to Case 3 (Figure 3.6).

Case 8: $q \in \sigma(p,t)$, e is a boundary of $\mathcal{D}_t$, and q is a vertex of P (Figure 3.11)
This case is analogous to Case 5 (Figure 3.8).

Case 9: $q \in \sigma(p,t)$, e is a boundary of $\mathcal{D}_t$, and f is a boundary of $\mathcal{D}_t$ (Figure 3.12)
This case is analogous to Case 6 (Figure 3.9).

Case 10: $q \in \sigma(p,t)$, e is a boundary of $\mathcal{D}_t$, and f is a boundary of $\mathcal{D}_s$ (Figure 3.13)
Point s must lie above h_p as in p the normal property is not yet violated. Furthermore, the point of curvature v of σ at point q must lie below h_q. The shortest self-approaching q-s path π_{qs} must travel through v and because h_q touches this point, there is a normal h_v that is parallel to h_p. Point s cannot lie above h_p and below h_v at the same time, hence this construction cannot exist.

Case 11: $q \in \sigma(p,t)$, e is a straight line segment, and q a vertex of P (Figure 3.14)
The normal h_p touches $\sigma(p,t)$ at point q which is a vertex of P. The shortest self-approaching p-t path π_{pt} must intersect or touch h_p at least once in point q' that lies between p and q. By definition we know that $|pq'| \geq |vq'|$ for every point v of $\pi_{pt}(p,q')$. Therefore, $|pq| \geq |vq|$, i.e. every point of the subpath $\pi_{pt}(p,q')$ lies in a disc centered at q with radius $|pq|$ (Figure 3.15). Thus, the straight line segment e is not part of π_{pt} and e is a part of σ due to a boundary d of $\mathcal{D}_s$. Segment e can only be part of σ if there is a point $w \in d \subset \sigma$ with a normal h_w that is parallel to h_p and $\sigma(p,w)$ only contains straight line segments and boundaries of $\mathcal{D}_s$. Otherwise, at the end of e there is either a bend point where σ turns to the right or e is tangent to a boundary of $\mathcal{D}_t$. In either option, no self-approaching p-t path can exist. Since w lies on a boundary of $\mathcal{D}_s$, there must be a point w' on the shortest self-approaching w-s path π_{ws} touched by h_w. Therefore,

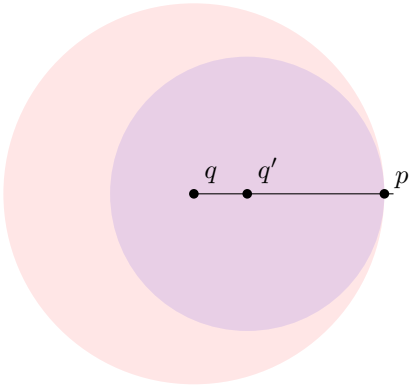


Figure 3.15: The disc with radius $|pq'|$ centered at point q' which lies on the straight line segment $\overline{pq}$ is fully contained in the disc with radius $|pq|$ centered at q.

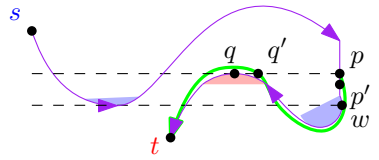


Figure 3.16: Point $q \in \sigma(p,t)$, p lies on a straight line segment (orange), and q lies on a boundary of $\mathcal{D}_t$ (red). The green path is the shortest self-approaching p-t path.

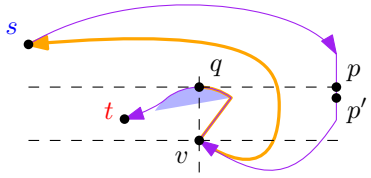
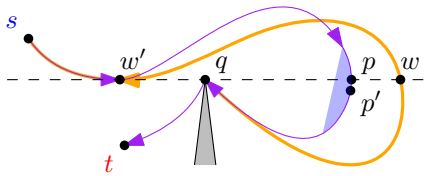


Figure 3.17: Point $q \in \sigma(p,t)$, p lies on a straight line segment (orange), and q lies on a boundary of $\mathcal{D}_s$ (blue). The orange path is the shortest self-approaching q-s path.

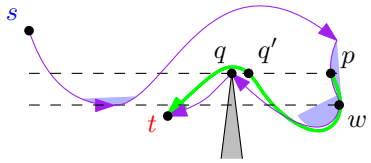
$\sigma(s,w)$ must intersect or touch h_w too. However, $\sigma(s,w)$ cannot intersect h_w as in h_p the normal property is not violated, thus σ cannot exist.

Case 12: $q \in \sigma(p,t)$, e is a straight line segment, and f a boundary of $\mathcal{D}_t$ (Figure 3.16)
This case is analogous to Case 11 (Figure 3.14).

Case 13: $q \in \sigma(p,t)$, e is a straight line segment, and f a boundary of $\mathcal{D}_s$ (Figure 3.17)
Point s must lie above h_p as in p the normal property is not yet violated. Furthermore, the center of curvature v of σ at point q must lie below h_p. The shortest self-approaching q-s path π_{qs} must travel through v and because h_q touches this point, there is a normal h_v that is parallel to h_p. Point s cannot lie above h_p and below h_v at the same time, hence this construction cannot exist.



(a) The point of curvature of σ at p lies in the direction of q. The orange path is the shortest self-approaching q-s path.



(b) Point of curvature of σ at p lies in the opposite direction of q. The green path is the shortest self-approaching p-t path.

Figure 3.18: Point $q \in \sigma(p,t)$, q is a vertex of P, and p lies on a boundary of $\mathcal{D}_s$ (blue).

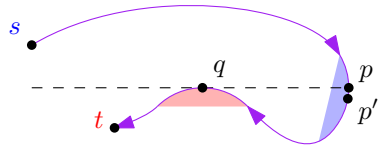


Figure 3.19: Point $q \in \sigma(p,t)$, q lies on a boundary of $\mathcal{D}_t$ (red), and p lies on a boundary of $\mathcal{D}_s$ (blue).

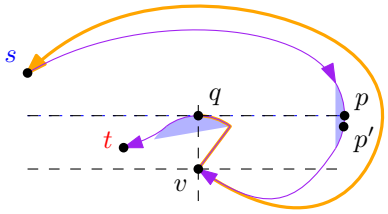


Figure 3.20: Point $q \in \sigma(p,t)$, q lies on a boundary of $\mathcal{D}_s$ (blue), and p lies on a boundary of $\mathcal{D}_s$ (blue). The orange path is the shortest self-approaching q-s path.

Case 14: $q \in \sigma(p,t)$, e is a boundary of $\mathcal{D}_s$, and q is a vertex of P (Figure 3.18)
There are two cases that need to be considered, either the point v of σ at point p lies in the direction of q with respect to p or not. First we consider the case where v lies in the direction of q (Figure 3.18a). Since q lies in the negative half-plane $h_{p'}^-$, the center of curvature of σ at point p must lie to the left of q. Therefore, the shortest self-approaching q-s path π_{qs} must first intersect or touch h_p to the right of q at point w and later to the left at point w'. Thus, $|qw'| < |ww'|$ which contradicts the self-approaching property of π_{vs}.

The case where v does not lie in the direction of q with respect to p (Figure 3.18b) is analogous to Case 11 (Figure 3.14).

Case 15: $q \in \sigma(p,t)$, e is a boundary of $\mathcal{D}_s$, and f is a boundary of $\mathcal{D}_t$ (Figure 3.19)
This case is analogous to Case 14 (Figure 3.18).

Case 16: $q \in \sigma(p,t)$, e is a boundary of $\mathcal{D}_s$, and f is a boundary of $\mathcal{D}_s$ (Figure 3.20)
Point s must lie above h_p as in p the normal property is not yet violated. Furthermore, the point of curvature v of σ at point q must lie below h_q. The shortest self-approaching q-s path π_{qs} must travel through v and because h_q touches this point, there is a normal h_v that is parallel to h_p. Point s cannot lie above h_p and below h_v at the same time, hence this construction cannot exist. □

3.1 An algorithm for paths with increasing chords

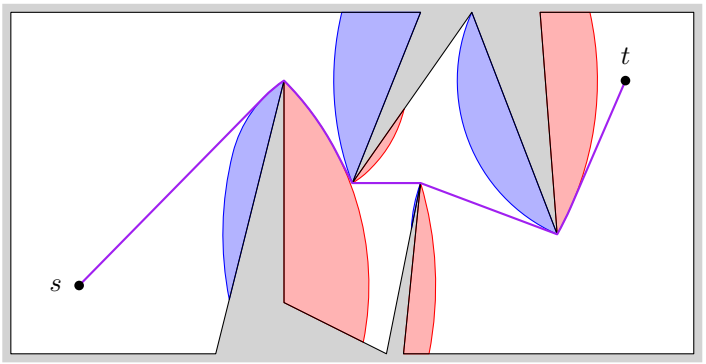
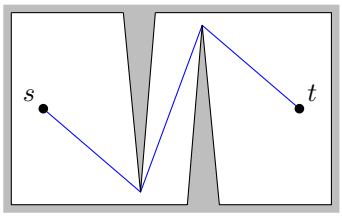
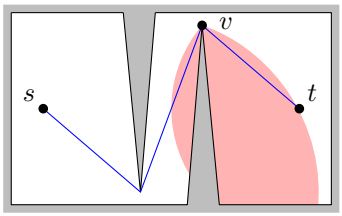


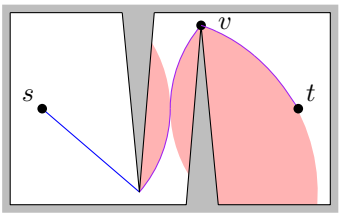
Figure 3.21: Polygon P with dead regions $\mathcal{D}_s$ (red) and $\mathcal{D}_t$ (blue), the geodesic (purple) of the remaining area is the shortest path with increasing chords.



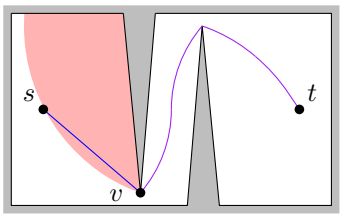
(a) The s-t geodesic σ inside P.



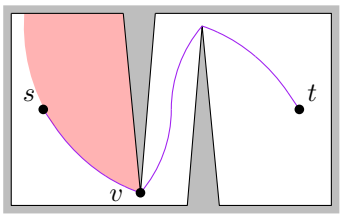
(b) Path σ goes through a dead region before and after vertex v.



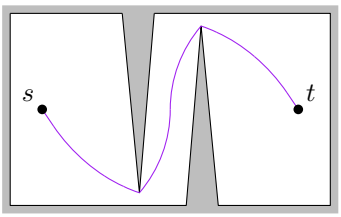
(c) The sections of σ before and after v are updated to avoid the dead regions.



(d) Path σ still goes through a dead region before new vertex v.



(e) The section of σ before v is updated to avoid the dead regions.



(f) No vertex v exist such that γ goes trough a dead region before or after v, thus γ is the shortest s-t path with increasing chords.

Figure 3.22: The calculation of the shortest s-t path with increasing chords inside a simple polygon P.

In this section we propose an algorithm that computes the shortest path with increasing chords. As discussed in Chapter 2 whenever a path contains involutes of an order higher than 1, the exact path cannot be calculated. However, we will assume it is possible to solve the equations of involutes of a higher order. First we will give an overview of the algorithm and, subsequently, go over the steps of the algorithm in more detail.

1. Construct all dead regions for s and t and subtract these from P.
2. Find the geodesic σ in $P \setminus (\mathcal{D}_t \cup \mathcal{D}_s)$, report σ.

By Theorem 13 we know that σ will be the shortest self-approaching s-t path in P. Figure 3.21 shows an example of a shortest s-t path with increasing chords inside a simple polygon P.

Constructing dead regions We can construct the dead regions as described by Bose et al. [5]. The construction of the dead regions is also described in Chapter 2.

Geodesic Finding the geodesic in polygon P without the dead regions $\mathcal{D}_t$ and $\mathcal{D}_f$ is almost analogous to finding the shortest self-approaching path π as described by Bose et al. [5]. The self-approaching algorithm starts with the s-t geodesic γ' and moving backward from t to s, the algorithm updates γ' if some section of γ' passes through the dead region $\mathcal{D}_t$. Whilst this does not immediately work for paths with increasing chords we can do something very similar. Let σ be the s-t geodesic in polygon P (Figure 3.22a). Select any vertex v of P that is in σ and update σ if it passes through one of the dead regions right before or after vertex v such that these dead regions are avoided (Figure 3.22b). The self-approaching algorithm describes how this step can be calculated. The only additional step that might need to be performed is calculating the bitangent between two dead regions of $\mathcal{D}_t$ and $\mathcal{D}_s$, for which we can use the bitangent calculation of the self-approaching algorithm [5]. Next, we mark vertex v as processed and choose new vertex v of P that is in σ and not yet marked as being processed. Thereafter, repeat the aforementioned steps (Figures 3.22d and 3.22e) until no vertices exist in σ that have not been processed. If at some

point σ cannot be updated or no bitangent exists, return that no s-t path with increasing chords exists. If all vertices in σ are marked as being processed we can return σ which is the shortest s-t path with increasing chords (Figure 3.22f).

Theorem 14. *The algorithm finds the shortest s-t path with increasing chords in a simple polygon P in $\mathcal{O}(n^2 \cdot g(n) + n^2 \cdot f(n))$, where $f(k)$ and $g(k)$ are the number of steps required to solve and evaluate, respectively, the equations of an involute of order k.*

Proof. The algorithm constructs a dead region for every internal node of SPT_s and SPT_t. By Lemma 8 this union of these dead regions is equal to $\mathcal{D}_s \cup \mathcal{D}_t$. Thereafter, the algorithm simply constructs the geodesic around the dead regions $\mathcal{D}_t$ and $\mathcal{D}_s$. By Theorem 13, this geodesic constructed is the shortest self-approaching path.

In the worst case, whilst computing the shortest self-approaching s-t path the entire set of dead regions $\mathcal{D}_t$ and $\mathcal{D}_s$ must be calculated. Given a polygon P with n vertices, the running time complexity of the shortest self-approaching path algorithm is $\mathcal{O}(n \cdot g(k) + n \cdot f(k))$, where k is the order of the highest involute that appears in the solution. Bose et al. [5] showed that k can be bound to n in a simple polygon. Therefore, computing all dead regions has a running time complexity that is similar to the complexity of the shortest self-approaching path algorithm: $\mathcal{O}(n \cdot g(n) + n \cdot f(n))$.

The update step of the geodesic can occur at most n times, as that is an upper bound on the number of vertices that can be marked as processed. The update step entails running the shortest self-approaching path algorithm ($\mathcal{O}(n \cdot g(n) + n \cdot f(n))$) and an additional bitangent calculation ($O(g(k) \log n + f(k))$).

Therefore, combining all these running times, our algorithm finds the shortest s-t path with increasing chords in $\mathcal{O}(n^2 \cdot g(n) + n^2 \cdot f(n))$. □

With this algorithm we conclude this chapter. We have proven that the shortest s-t path with increasing chords is unique in a simple polygon. Moreover, the shortest s-t path with increasing chords is the s-t geodesic in polygon P without the dead regions $\mathcal{D}_t$ and $\mathcal{D}_s$. Finally, we have shown that the shortest s-t path with increasing chords can be calculated using the shortest self-approaching path algorithm.

Chapter 4

Self-Approaching Paths in Polygonal Domains with Holes

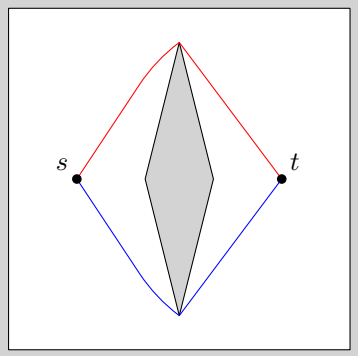


Figure 4.1: Two distinct shortest self-approaching s-t paths.

The algorithm for the shortest self-approaching s-t path only works for simple polygons. A logical next step is therefore to try and calculate the shortest self-approaching s-t path in a polygon with holes. In a simple polygon there exists only one unique shortest self-approaching path [5]. In a polygon with holes, however, multiple shortest self-approaching paths can coexist as is shown in Figure 4.1. In this chapter we will first give a number of examples that show that finding the shortest self-approaching path in a polygon with holes is not trivial. Furthermore, we will give an algorithm that can compute a shortest self-approaching s-t path polygon with holes, admittedly the running time of this algorithm is unbounded.

4.1 Exploration for trivial self-approaching algorithms

First, we will explore some polygons in which finding the shortest self-approaching path is not as trivial as might be initially expected.

4.1.1 Shortest geodesic

An intuitive idea for finding a self-approaching path, would be to construct a self-approaching path that is homotopically equivalent to the 1-geodesic, i.e. the shortest s-t path. However, Figure 4.2 shows a polygon where there is no self-approaching path that is homotopically equivalent to the 1-geodesic. However, there exists a self-approaching path that is homotopically equivalent to the 2-geodesic.

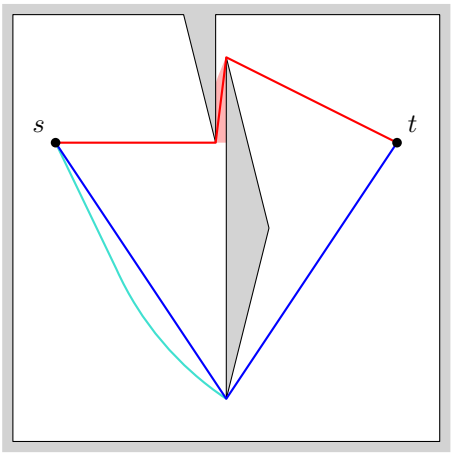


Figure 4.2: There is no self-approaching path that is homotopically equivalent to the 1-geodesic (red). However, for the 2-geodesic (blue) a self-approaching path (turquoise) exists that is homotopically equivalent.

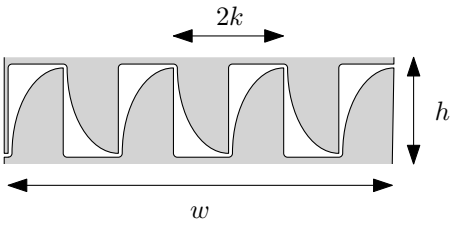


Figure 4.3: Rectangular section of a polygon where white is the only path through this rectangle. The path is constructed out of cycloids generated by circles with diameter k, such that $k = 2h/\pi$.

4.1.2 Shortest geodesic with self-approaching path

Even if we can find a k-geodesic, for smallest possible k, where a self-approaching path π exist that is homotopically equivalent, there is no guarantee that π would actually be the shortest self-approaching path in the polygon. First we introduce a *jagged* structure as described by Icking et al. [13] and which is depicted in Figure 4.3. The jagged structure lies in a w by h rectangle and is constructed out of pieces of cycloids that are generated by circles of diameter k, where $k = 2h/\pi$. With this choice of k the length of a path around one piece of a cycloid is exactly $2k$. Therefore, not only is a path going through a jagged structure self-approaching, the length of this path is exactly $2w$. Furthermore, if we let the number of cycloids approach infinity, h approaches 0 which makes this structure particularly useful for our example.

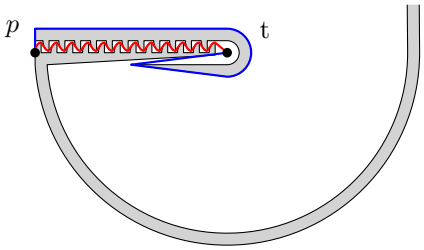
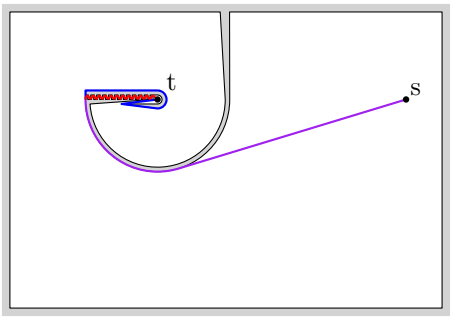
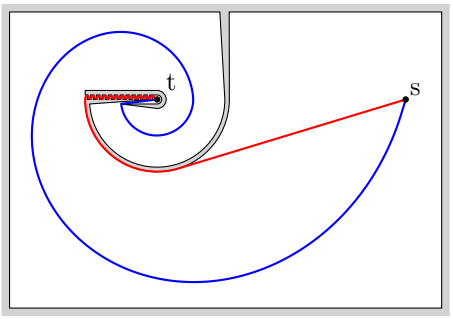


Figure 4.4: Point t lies enclosed in a hole, such that both geodesics connecting p and t that go around the hole (blue) and trough the jagged section (red) are approximately of the same length.



(a) The s-t 1-geodesic γ_1 (blue) and the s-t 2-geodesic γ_2 (red), the part where γ_1 and γ_2 overlap is in purple.



(b) The self-approaching s-t paths π_1 and π_2 that are homotopically equivalent to γ_1 and γ_2, respectively. Path π_1 is significantly longer than π_2.

Figure 4.5: Polygon P such that the self-approaching s-t path is not homotopically equivalent to the s-t 1-geodesic.

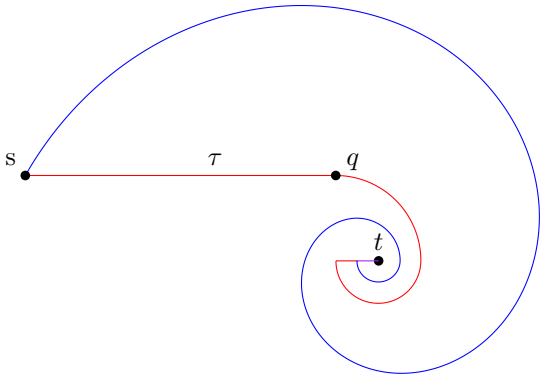


Figure 4.6: Path π_2 (red) and path π_1 (blue), point s is the intersection of the involute of π_1 and the tangent τ to the involute of π_2 at point q.

We aim to construct a polygon P where both a k-geodesic γ_1 and a k'-geodesic γ_2 exist, such that $k < k'$. Furthermore, for both geodesics a homotopically equivalent self-approaching path must exist and the shortest self-approaching path π_1 equivalent to the k-geodesic is longer than the shortest self-approaching path π_2 equivalent to the k'-geodesic. We try to make $|\pi_1|/|\pi_2|$ as large as possible. This ratio tells us that if there exists an algorithm that can find the k-geodesic, such that k is minimal and there exists a self-approaching path homotopically equivalent to the k-geodesic, this algorithm is at least a $|\pi_1|/|\pi_2|$-approximation.

Figure 4.4 shows an exaggerated view of our construction of P. Let point p be at the entrance of the jagged structure and t at the exit, furthermore let the Euclidean distance between p and t equal 2 (Figure 4.4). If the number of cycloids approaches infinity and the thickness of the hole approaches 0, the path connecting p and t going through the jagged structure approaches length $4 + \epsilon$, for arbitrary small ϵ. Furthermore, the path going around the jagged structure also approaches a length of 4. We can then choose point s such that the s-t 1-geodesic γ_1 and the s-t 2-geodesic γ_2 both go through point p and $\gamma_1(s, p) = \gamma_2(s, p)$. Therefore, γ_1 will go around the hole and γ_2 goes through the jagged section (Figure 4.5a). Moreover, we want that there exist self-approaching s-t paths π_1 and π_2 that are homotopically equivalent to γ_1 and γ_2, and such that π_1 is longer than π_2. ((Figure 4.5b)).

With the hypothetical structure of polygon P, we look for the point s such that the ratio between π_1 and π_2 is as high as possible. To compute this ratio, a grid search is performed on the location of s. Path π_2 is made out of the jagged structure followed by an involute I. We construct

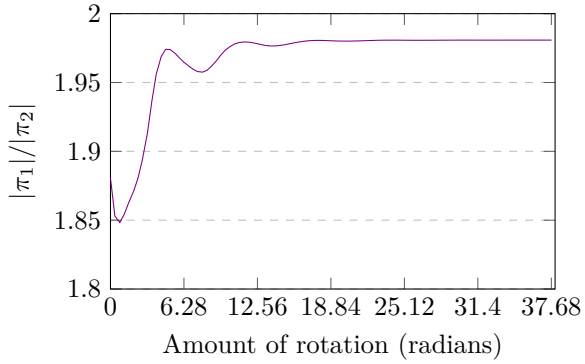


Figure 4.7: The ratio between the lengths of two self-approaching s-t paths π_1 and π_2 such that the geodesic homotopically equal to π_1 is shorter than the geodesic homotopically equal to π_2.

I piecewise, after every piece we compute the tangent τ to the involute I at point q (Figure 4.6). As we know exactly at which angles the segments of I start and end, we can calculate the exact length of I constructed so far. Thereafter, we need to know where τ intersects with π_1. As π_1 is also constructed of involutes of a high order it is not possible to find this intersection analytically. Therefore, a binary search is applied in order to find the intersection of τ and π_1 with relative high accuracy. Once the intersection point is determined we have a clear definition of π_1 and π_2 and we can calculate their lengths by integrating their individual segments. The results obtained by the script given in Appendix A shows that if we continuously increase the size of π_2, $|\pi_1|/|\pi_2| \approx 1.98$. Figure 4.7 shows the progression of the ratio between π_1 and π_2 with respect to how much τ has rotated during construction, i.e. if τ made a full rotation, we say that we have constructed approximately $2 \cdot 3.1415$ of I.

4.2 Self-approaching algorithm for holes

The algorithm presented in this section returns a shortest self-approaching s-t path in a polygon P with holes, given that such a path exists. Whilst this path returned by the algorithm is indeed the a shortest self-approaching path, the algorithm is not efficient. Before we propose our new algorithm, we need to generalize the following lemma that was given by Bose et al. [5] for simple polygons. The lemma can simply be lifted from its simple polygon constraint without any modifications.

Lemma 15. *Bends of a shortest self-approaching path π in a polygon P form a subset of vertices of P.*

As we have shown in Section 4.1 an algorithm for finding the shortest self-approaching s-t path is not as trivial as one might expect. Therefore, this algorithm explores all possible shortest self-approaching paths whilst ignoring subpaths that will not contribute to the output. The idea of this algorithm is to simulate a wavefront originating from t that expands in all regions for which a self-approaching path to t exists, as was inspired by the work of Hershberger et al. [11]. Once the wavefront reaches s we have found a shortest self-approaching s-t path. Let V be the set of vertices of P and vertex s. We will first give a high level overview of the algorithm:

1. The algorithm first finds every vertex $v \in V$ that is directly visible from t and stores the line segments $\overline{vt}$ in a priority queue Q.
2. This step will expand the shortest path π constructed so far to every vertex $w \in V$. Therefore, we are guaranteed to explore every possible path going to t, and eventually explore a shortest self-approaching s-t path. More specifically, we test if we can append π by a straight line segment, involute, or involute and straight line segment obtaining π^* such that π^* is a self-approaching w-t path. If π^* exists without intersecting with P, this path can be added to Q.

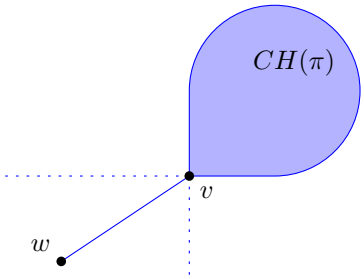


Figure 4.8: The case such that $\overline{vw} \oplus \pi$ is a self-approaching w-t path.

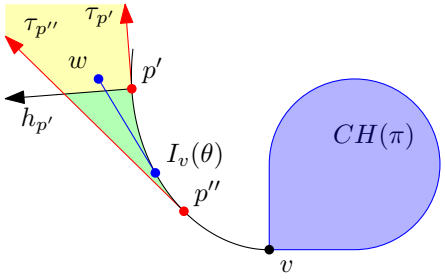


Figure 4.9: The case such that $\overline{wI_v(\theta)} \oplus I_v(\theta, v) \oplus \pi$ is a self-approaching w-t path. The green region lies to the right of $\tau_{p''}$, the left of $h_{p'}$, and outside the dead region of $CH(\pi)$. The yellow region lies to the right of $\tau_{p''}$ and $h_{p'}$, and to the left of $\tau_{p'}$.

3. Next, we want to ignore all paths that will not be part of the shortest self-approaching s-t path π_{st}. Let π_1 and π_2 be two paths in Q such that π_1 and π_2 both start in the same vertex, π_1 is shorter than π_2, and the convex hull of π_1 lies in the convex hull of π_2. Path π_2 could never be part of π_{st} as π_1 can always replace π_2 and make the resulting path shorter. Therefore, if π_1 and π_2 both exist in Q, π_2 can be removed.
4. Thereafter, we repeat Steps 2 and 3 until π is an s-t path or Q gets empty. If π is an s-t path the algorithm returns π, otherwise, the algorithm reports that no self-approaching s-t path exists.

The algorithm can be divided into three sections: initialization (1), propagation (Steps 2 and 3), and termination (Step 4). We will now provide a more detailed overview of the algorithm.

Initialization Let priority queue Q store self-approaching v-t paths for arbitrary vertex v, sorted on the length of these paths in increasing order. Furthermore, let SAPM be a self-approaching path map that holds a set of self-approaching paths connecting arbitrary vertex w to t. All vertices that are directly visible by t can reach t by a self-approaching path, which is a simple straight line. These paths are added to SAPM and Q.

Propagation Let π be the path that is at the front of Q which starts in vertex v, remove π from Q and SAPM(v). We loop over every vertex $w \in V$ that is not in the shadow of v with respect to π. If the angle between $\overline{vw}$ and the tangents to $CH(\pi)$ at v in both directions are less than 90°, we need no involute to connect to w as is shown in Figure 4.8. If $\overline{wv}$ does not intersect with P we append $\overline{wv}$ to π, obtaining path π^*. Path π^* can be added to SAPM(w) and Q.

If the angle between $\overline{wv}$ and the tangent to $CH(\pi)$ in both directions is not less than 90° or $\overline{wv}$ intersects with P, we cannot append π with a straight line segment (Figure 4.9). When this is the case we test is we can append π with an involute of $CH(\pi)$ and a straight line segment to this involute that does not intersect with P. First we test if we can append π with a clockwise involute and thereafter if we can append π with a counterclockwise involute. We start by constructing the

clockwise involute I_v of $CH(\pi)$ that starts in v piece by piece. We keep constructing I_v until we find segment $I_v(p', p'')$, where the curve orientates from p' to p''. Let $\tau_{p'}$ and $\tau_{p''}$ be rays that are in the opposite direction of the tangent to I_v at p' and p'', respectively (Figure 4.9). Furthermore, $h_{p'}$ is the ray perpendicular to $\tau_{p'}$ which points outwards. We keep constructing I_v until we find segment $I_v(p', p'')$, such that w lies to the right of $\tau_{p''}$ and either w lies to the left of $h_{p'}$ or to the right of $h_{p'}$ and to the left of $\tau_{p'}$ (Figure 4.9). Thereafter, we find the line segment $\overline{qw}$ between $I_v(p', p'')$ and w, such that $\overline{qw}$ is tangent to I_v at point q. If $\overline{qw}$ does not intersect a boundary of P we can extend π to π^* such that $\pi^* = \overline{wq} \oplus I_v(q, v) \oplus \pi$, otherwise we move to the next segment of I_v. We can stop the construction of I_v if either $I_v(p', p'')$ intersects with a segment of P or we have found a proper tangent $\overline{wI_v(\theta)}$. If π^* exists we can add it to SAPM(w) and inserted it into Q. Finally we take analogous steps in computing the counterclockwise involute and extending path π into π^*.

Thereafter, we check in SAPM(w) whether it is possible to remove a path from Q and SAPM(w). Assume there are two paths π_1 and π_2 in SAPM(w) such that $|\pi_1| \leq |\pi_2|$ and $CH(\pi_1) \subseteq CH(\pi_2)$, i.e. π_1 is not longer than π_2 and the convex hull of π_1 is contained in the convex hull of π_2. Then we remove π_2 from both Q and SAPM(w).

Termination Whenever path π, taken from the front of Q, is a path going from s to t the algorithm can terminate and report π as a shortest path. If Q gets empty at some point, no self-approaching path exist connecting s to t which can be returned.

Running time The proposed algorithm is greatly inefficient as the running time cannot be bounded in terms of the input or output. Even if the algorithm could have been bounded in terms of the input or output we still could have to search trough a number of different paths that is equal to the number of k-geodesics that exist in P which is at least exponential in terms of the number of holes. Moreover, as these self-approaching paths again contain involutes, we have to solve and evaluate their equations which is not possible analytically. Therefore, the running time analysis is omitted and the algorithm should be seen as a proof of concept which could be improved upon in the future.

Theorem 16. *The algorithm finds a shortest self-approaching s-t path in a polygon with holes P.*

Proof. The proposed algorithm is a fairly trivial algorithm that tests all paths possible. The removing of paths is safe, as the w-t path π_2 is only removed if a w-t path π_1 exists such that the convex hull of π_1 is contained in the convex hull of π_2 and length of π_1 is less or equal than the length of π_2. Therefore, for any extension α such that $\alpha \oplus \pi_2$ is self-approaching, $\alpha \oplus \pi_1$ must also be self-approaching. Furthermore, the length of $\alpha \oplus \pi_2$ cannot be longer than $\alpha \oplus \pi_1$. Hence, it suffices to only take π_1 into account remove π_2.

The algorithm terminates once the path π at the front of priority queue Q is an s-t path. Path π is the shortest self-approaching s-t path in polygon P. If there is a shorter self-approaching path π', then this path cannot be an extension of any self-approaching path ending in a vertex v extended with a straight line segment, involute, or involute and straight line segment as such an option would have been explored by the algorithm. Therefore, π' must be a straight line segment or have a bend point that is no vertex of P. If π' was a straight line segment the path would have been found in the initialization step and π would not have been the first s-t path in Q. Furthermore, π' cannot have a bend point that is no vertex of P by Lemma 15. Hence, π is the shortest self-approaching s-t path in P. □

Chapter 5

Characterization of Strongly Connected Polygons

A polygon P is *strongly connected* if and only if any two points in P can be connected by a constrained path. The curve constrained paths we are interested in for this chapter are again self-approaching paths and paths that have increasing chords. Bose et al. [5] described an algorithm that is able to determine if a simple polygon is strongly connected in terms of self-approaching paths and paths with increasing chords. However, Stefan Langerman proposed the question how to characterize polygons with holes that are strongly connected.

In this chapter we will look at how holes influence strongly connected polygons and characterize the shape of holes when they are part of a strongly connected polygon. Furthermore, we will show that a polygon can be determined to be strongly connected in case of self-approaching paths and paths with increasing chords by first ignoring all holes and testing the resulting simple polygon and thereafter by separately testing if the holes match our characterizations.

5.1 Self-approaching polygons

A *self-approaching polygon* P is a polygon where for any points $s, t \in P$ a self-approaching s-t path exists. In this section we will characterize how the shapes of holes can be characterized when they are part of a self-approaching polygon. A *self-approaching hole* H is a hole such that for any points $s, t \in U \setminus H$, where U is an infinitely large convex polygon completely containing H, a self-approaching s-t path exists. Bose et al. proved Theorem 17, which will be used often in the following proofs.

Theorem 17 ([5])**.** *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 exactly one connected component.*

First we will prove that if any number of self-approaching holes are added to a self-approaching polygon the polygon will remain self-approaching. Therefore, we can later solely focus on characterizing the shape of a self-approaching hole without taking into account the polygon this hole lies in.

Lemma 18. *Polygon P is self-approaching if and only if every hole H in P is self-approaching and the simple polygon P', that is defined by the outer boundary of P, is self-approaching.*

Proof. Assume there is a self-approaching polygon P, but there is some hole H in P that is not self-approaching or simple polygon P' is not self-approaching. If there is a hole H that is not self-approaching, there is a disc D centered at point $p \in U \setminus H$ such that $D \cap (U \setminus H)$ has more than one connected component (Figure 5.1). Let q be a point that lies in a different connected component than p. For every disc D' centered at p' that lies on segment $\overline{pq}$ and radius $|p'q|$. $D' \cap P$

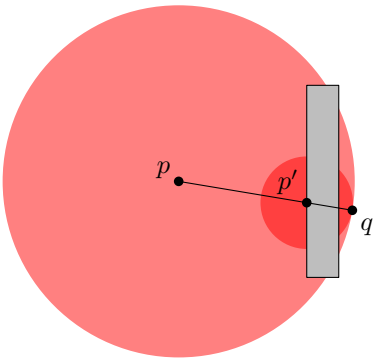


Figure 5.1: If a disc centered at point p that has more than one connected component, the disc centered at p' such that $p' \in \overline{pq}$ and with radius $|pq|$ also has more than one connected component.

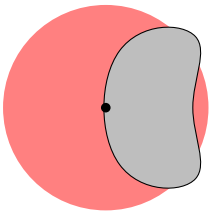


Figure 5.2: A concave hole (grey) that separates a disk (red) into two connected components.

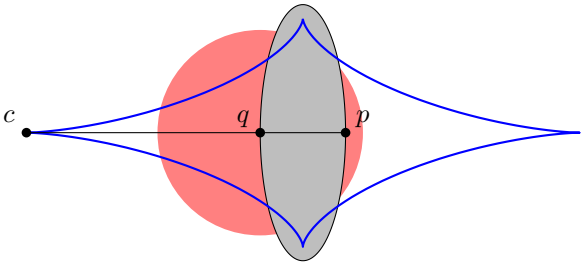


Figure 5.3: A hole (grey) with a partially external evolute (blue) and a disk (red) that is separated into two connected components.

will also have more than one connected component. Therefore, since H is interior to P there must be a disc D' for some point $p' \in P$ with radius $|p'q|$, such that $D' \cap P$ has more than one connected component. Hence P cannot be self-approaching. Furthermore, an analogous contradiction can be reached for when P' is not self-approaching. Thus, if P is self-approaching, every hole H in P and simple polygon P' are self-approaching.

Now assume instead that every hole in P and simple polygon P' are self-approaching, but polygon P is not self-approaching. There must be a disk D centered at point $p \in P$, such that $D \cap P$ has more than one connected component. Neither a hole in P or the boundary of P' can divide D in more than one connected component. Disc D can only exist if two holes or a hole and the boundary of P' intersect. However, this is not allowed as that would mean that the two holes would actually be one hole or the hole is not interior to P. Hence, P must be self-approaching. □

With Lemma 18 we can now solely focus on the shape of H without considering the surrounding polygon. Recall that the *evolute* of a curve is the locus of all centers of curvatures of the curve, Figure 5.3 shows the evolute of an ellipse.

Theorem 19. *A hole H is self-approaching if and only if the boundary is convex and the evolute of the boundary is entirely contained inside H.*

Proof. If the boundary of H is not convex, it is easy to see that such a hole is not self-approaching. A disc centered across the concave part clearly does not have one connected component, an example is shown in Figure 5.2.

If the evolute is not fully contained in H, consider point p on the boundary of H for which the center of its curvature c lies outside of H (Figure 5.3). Let disk D be centered at point q somewhere on line segment $\overline{cp}$ but still outside H with a radius of $|qp| + \epsilon$ for small ϵ. Because

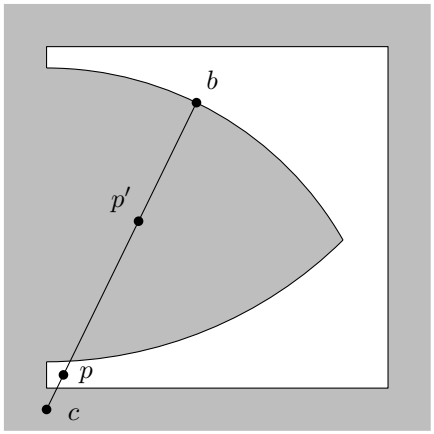


Figure 5.4: A boundary defined by curves and arcs that is not self-approaching. Point c is the center of curvature to the boundary at point b.

D has a smaller curvature than the section p lies on, $D \setminus H$ must have more than one connected component and hence, H is not self-approaching.

Next, assume H is not self-approaching, but is convex and the evolute is contained inside H. Then, there must be some disc D centered outside H such that $D \setminus H$ is divided into two connected components. Therefore, the boundary of D must intersect the boundary of H at least twice. H is convex, thus there must be some section of H with a curvature that is larger than D. Since D is centered outside of H, the center of curvature from this section must lie even farther outside of H and therefore the evolute cannot be contained inside H. □

Therefore, if one desires to test if a polygon with holes is self-approaching, they can simply test the polygon whilst ignoring the holes using the algorithm proposed by Bose et al. [5] and test if every hole is convex and the evolute is internal to the hole. However, a polygon with holes cannot be self-approaching if the holes are defined by straight line segments. Hence, holes must be defined by arcs and curves instead of straight line segments. If holes can be defined by arcs and curves, there is no reason to not allow this for the outer boundary of the polygon. Even though the algorithm proposed by Bose et al. operates on a polygon with vertices and straight line segments, testing if a polygon B defined by curves and arcs is self-approaching is analogous to testing if a hole is self-approaching (Lemma 20).

Lemma 20. *Polygon B that is defined by curves and arcs instead of straight line segments is self-approaching if and only if for every point b on the boundary of B with center of curvature c, no points p and p' exist such that both points lie on segment $\overline{bc}$, p lies inside B, p' lies outside B, and p' lies between p and b (Figure 5.4).*

Proof. This proof is analogous to the proof of Theorem 19. □

5.2 Polygons with increasing chords

In the previous subsection we characterized the shapes of holes that lie inside a self-approaching polygon. The natural next question is to characterize the shapes of holes that lie in polygons with increasing chords. A *polygon with increasing chords* P is a polygon where for all points $s, t \in P$ an s-t path with increasing chords exists. We say that a *hole with increasing chords* H is a hole that for every points $s, t \in U \setminus H$, where U is again an infinitely large convex polygon, a path with increasing chords exists.

First we will prove Lemma 21 using similar techniques that Bose et al. [5] used for proving part of Theorem 17. A *vesica piscis* V_{pq} is the intersection of discs D_p and D_q centered at points p and q with radius $|pq|$.

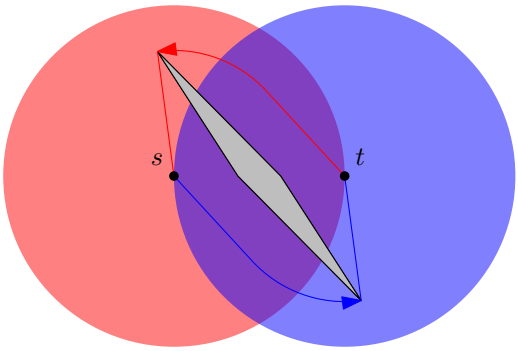


Figure 5.5: A self-approaching s-t path (blue) and self-approaching t-s path (red) exist, but no s-t path with increasing chords exists.

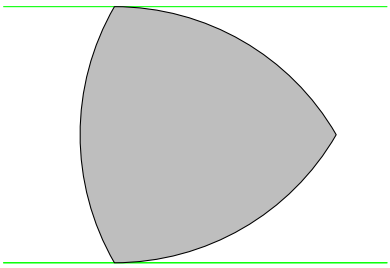


Figure 5.6: Two supporting lines (green) of a shape with constant width (gray).

Lemma 21. *If polygon P has increasing chords, then any points $p, q \in P$ must lie in the same connected component of $V_{pq} \cap P$, where V_{pq} is the vesica piscis defined by p and q.*

Proof. Assume that P has increasing chords, but there exists p and q such that they do not lie in the same connected component in $V_{pq} \cap P$. If p and q are not in the same connected component, any path γ connecting p and q must exit V_{pq} at some point. Let r be a point on γ that lies outside V_{pq}. Either $|pq| < |pr|$ or $|pq| < |qr|$, this inequality always contradicts the increasing chords property of P.

Now assume that P does not have increasing chords and for any points $p, q \in P$, p and q lie in the same connected component of $V_{pq} \cap P$. Therefore, there exists points $s, t \in P$ that cannot be connected by a path with increasing chords. However, we know that P is self-approaching. Otherwise there is some disc D centered at $p' \in P$ such that $D \cap P$ has two connected components. If we then choose q' to be in a different connected component than p', clearly p' and q' do not lie in the same connected component in $V_{p'q'} \cap P$. Hence, for points s and t both a self-approaching s-t path π_{st} and a self-approaching t-s path π_{ts} exist. Recall that a path with increasing chords is self-approaching in both directions. Therefore, π_{st} must leave the disc with radius $|st|$ that is centered at s and π_{ts} must leave the disc with radius $|st|$ that is centered at t on different sides of the line passing through s and t (Figure 5.5). However, then the points s and t do not lie in the same connected component of $V_{st} \cap P$, resulting in a contradiction. □

As with self-approaching polygons we can say that holes with increasing chords can be added to polygons with increasing chords without violating the increasing chords property.

Lemma 22. *Polygon P has increasing chords if and only if every hole H in P has increasing chords and the simple polygon P' that is obtained by filling in the holes of P has increasing chords.*

Proof. This proof is analogous to the proof of Lemma 18. □

With Lemma 22 we can now again focus on the characterization of the shape of hole. If we know all holes in polygon P have increasing chords and that P without holes has increasing chords, P also has increasing chords. Before we characterize holes with increasing chords we require an extra definition: two distinct lines are *supporting lines* of a shape if they are parallel and touch, but never intersect, the shape (Figure 5.6).

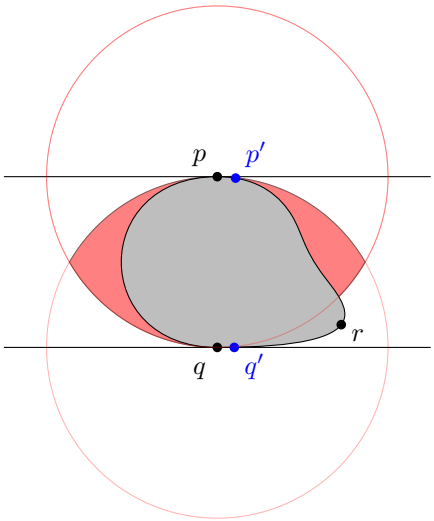


Figure 5.7: A hole (gray) that is not contained in vesica piscis V_{pq} and points p' and q' that lie in the ϵ-neighborhoods of p and q, respectively.

Lemma 23. *If a hole H has increasing chords then $H \subseteq V_{pq}$, where p and q are points touched by any pair of supporting lines of H and V_{pq} is a vesica piscis defined by p and q.*

Proof. Let H have increasing chords and assume supporting lines that touch H in p and q exist such that $H \not\subseteq V_{pq}$. Assume, w.l.o.g, that $H \not\subseteq D_p$, where D_p is a disc centered at p with radius $|pq|$. Consider the points p' and q' that lie in the ϵ-neighborhoods of p and q and towards point r that lies on the boundary of H outside D_p (Figure 5.7). Since H cannot have straight lines and p and q touch the supporting lines, $|p'q'| < |pq'|$. Furthermore, since p' and q' lie in the ϵ-neighborhood of p and q respectively, $|p'q'| < |p'r|$. Since p and r lie on opposite sides of the line going through p' and q', p' and q' lie in two different connected components in $V_{p'q'} \cap (U \setminus H)$, where $V_{p'q'}$ is the vesica piscis defined by p' and q', and U is an infinitely large convex polygon that fully contains H. Therefore, by Lemma 21 hole H cannot have increasing chords. □

Using Lemma 23 we can prove Theorem 24 that shows a hole has increasing chords if and only if the shape of the hole has constant width. A shape is of *constant width* if and only if the distance between every pair of supporting lines is equal (Figure 5.6). Furthermore, a shape is of constant width if and only if all supporting lines are orthogonal to the line connecting the points touched by these supporting lines [20].

Theorem 24. *Hole H has increasing chords if and only if the shape of H has constant width.*

Proof. Assume that H is of constant width but does not have increasing chords. By definition, a supporting line cannot touch H twice, therefore H is convex and does not contain straight-line segments. If H does not have increasing chords, there exist points p and q outside of H that lie not in the same connected component of vesica piscis V_{pq} that is defined by p and q. We can therefore select point p' and q' that lie on segment $\overline{pq}$ and on the boundary of H (Figure 5.8). Therefore, the distance between the supporting lines that are perpendicular to the segment $\overline{pq}$ is larger than $|p'q'|$. However, since H is convex when the supporting lines are rotated towards p' and q' the distance between the lines must decrease. Therefore, H cannot be of constant width.

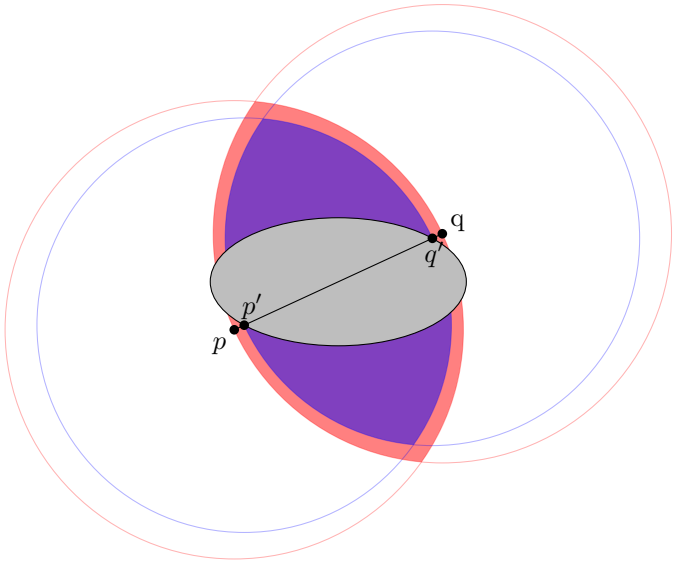
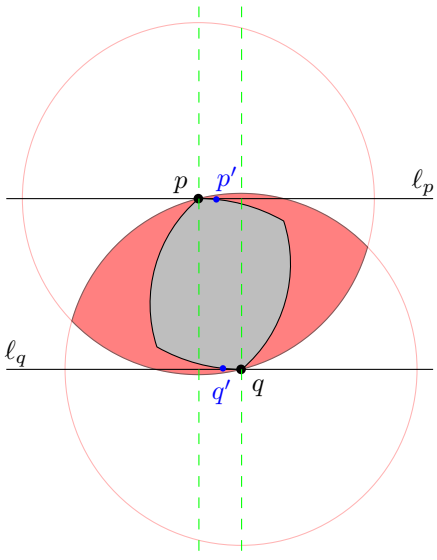
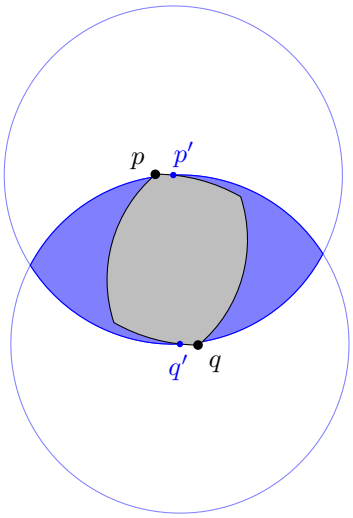


Figure 5.8: If p and q do not lie in the same connected component in $V_{pq} \cap P$, p' and q' also do not lie in the same connected component in $V_{p'q'}$.



(a) p' and p' lie in the ϵ-neighborhoods of p and q respectively and between the perpendicular lines (green) through p and q.



(b) p' and q' do not lie in the same connected component in $V_{p'q'} \setminus H$ (blue).

Figure 5.9: A hole (gray) such that $\overline{pq} \not\perp \ell_p$.

Assume that H has increasing chords but is not of constant width. Let ℓ_p and ℓ_q be any supporting lines that touch H in p and q, but where $pq \not\perp \ell_p$, i.e. the line segment connecting p and q is not orthogonal to ℓ_p. Since H has increasing chords, by Lemma 23, $H \subseteq V_{pq}$ where V_{pq} is the vesica piscis defined by p and q. Furthermore, we know that H must lie in between ℓ_p and ℓ_q. Any hole that has increasing chords is also self-approaching by definition. Therefore, the centers of curvatures for H must be contained in H by Theorem 19. Assume, w.l.o.g., that ℓ_p is horizontal and p lies to the left of q (Figure 5.9a). Because H is contained in between ℓ_p and ℓ_q, the center of curvature for p must lie to the left of the perpendicular to ℓ_p that goes through p. Analogously, the center of curvature for q must lie to the right of the perpendicular to ℓ_q that goes through q. Let p' and q' be points that lie on the boundary of H, in the ϵ-neighborhood of p and q respectively, and between the two perpendiculars. Let $V_{p'q'}$ be the vesica piscis defined by p' and q'. Because of the curvature of H, p and q both fall outside $V_{p'q'}$ on opposite sides of a line going through p' and q', as is demonstrated in Figure 5.9b. Therefore, p' and q' are not in the same connected component in $V_{p'q'} \setminus H$ and by Lemma 21 hole H cannot have increasing chords.

Thus, Hole H has increasing chords if and only if the shape of the hole has constant width. □

Analogous to the self-approaching polygons, to determine whether a polygon P has increasing chords the problem can be divided into two subproblems. First we can test if the simple polygon P' defined by the outer boundary of P has increasing chords with the algorithm described by Bose et al. [5]. Secondly, the holes can be tested to be of constant with. Only if both tests are positive the original polygon has increasing chords. Of course, as with the self-approaching holes, holes with increasing chords must be defined by curves and arcs instead of straight line segments. We know that P' has increasing chords if P' is self-approaching [5]. Thus, if P' also is defined by curves and arcs, we can simply use Lemma 20 to test whether P' has increasing chords.

Chapter 6

Discussion

In this thesis two types of constrained paths are explored in different settings. We discussed how to construct a path with increasing chords in a simple polygon assuming that transcendental equations can be solved. Furthermore, we discussed how a self-approaching path in any polygonal domain can be constructed and why this is not a trivial task. Finally, we characterized the shapes of holes in polygons that are strongly connected in terms of self-approaching paths and paths with increasing chords.

In Chapter 2 we started by exploring the definitions of self-approaching paths and paths with increasing chords and how these types of paths are related. We discussed the behavior of shortest self-approaching paths in simple polygons and how these paths can be defined. Additionally, we introduced the notion of holes and what it means for paths to be shortest in their homotopy group.

In Chapter 3 we first explored some useful properties of paths with increasing chords in simple polygons. With these properties we were able to show that the shortest s-t path in a polygon minus the dead regions for self-approaching paths going to s and t is the shortest path with increasing chords. Therefore, the algorithm for finding the shortest path with increasing chords is analogous to finding a shortest self-approaching path.

Chapter 4 we discussed why some trivial algorithms for finding a self-approaching path in a polygon with holes might not work. Therefore, an algorithm is proposed that can find the shortest self-approaching path, albeit with an unbounded running time. This algorithm can act as a baseline for future research.

Finally, Chapter 5 characterizes the shape of self-approaching holes and holes with increasing chords. The evolute of the boundary of a self-approaching hole must be internal to the hole. Furthermore, the shape of a hole with increasing chords is always a shape of constant width. Additionally, we show that testing the holes separate from the polygon is sufficient for testing if a polygon with holes is strongly connected.

6.1 Future work

There are a number of interesting directions for work related to this thesis in the future. First of all, the question remains open for whether an efficient algorithm exists that can find a shortest self-approaching s-t path in a polygon with holes. Perhaps the algorithm proposed in Section 4 can be improved in having a polynomial bounded running time or the problem could be proven to be NP-hard.

Another natural next step could be to find an algorithm that finds a shortest s-t path with increasing chords in a polygon with holes. Furthermore, for both types of constrained paths an interesting direction could be in looking into settings where the number of dimensions exceeds two. Although, this will likely be hard as even finding the geodesic in a 3D space is NP-hard [6].

Another direction for future research could be in seeing if the results from this thesis can be applied on angle monotone constrained paths. Recall that an φ-self-approaching s-t path π^{φ} is a

path where for any point p, subpath $\pi^{\varphi}(p,t)$ falls entirely in a wedge with angle φ and apex p. An s-t φ-path with increasing chords is simply a path that is an φ-self-approaching s-t path and a φ-self-approaching t-s path. For example, it could be interesting to see the influence changing φ has on the characterization of the holes in strongly connected polygons. Similar questions can be asked for φ-increasing-chord paths, which are φ-self-approaching in both directions. Additionally, the stretching factor for φ-self-approaching paths is still unknown and could therefore be an interesting topic.

Bibliography

[1] O. Aichholzer, F. Aurenhammer, C. Icking, R. Klein, E. Langetepe, and G. Rote. Generalized self-approaching curves. *Discrete Applied Mathematics*, 109(1):3–24, 2001. 14th European Workshop on Computational Geometry.

[2] Patrizio Angelini, Fabrizio Frati, and Luca Grilli. An algorithm to construct greedy drawings of triangulations. In Ioannis G. Tollis and Maurizio Patrignani, editors, *Graph Drawing*, pages 26–37. Springer Berlin Heidelberg, 2009.

[3] M. Biro, J. Gao, J. Iwerks, I. Kostitsyna, and J.S.B. Mitchell. Beacon-based routing and coverage. In *21st Fall Workshop on Computational Geometry (FWCG 2011)*, 2011.

[4] M. Biro, J. Gao, J. Iwerks, I. Kostitsyna, and J.S.B. Mitchell. Combinatorics of beacon-based routing and coverage. In *Proceedings of the 25th Canadian Conference on Computational Geometry (CCCG)*, pages 1–6, 2013.

[5] P. Bose, I. Kostitsyna, and S. Langerman. Self-approaching paths in simple polygons. *Computational Geometry*, 87, April 2020.

[6] J. Canny and J. Reif. New lower bound techniques for robot motion planning problems. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 49–60, 1987.

[7] Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6(3):485–524, 1991.

[8] J. Gao and L. Guibas. Geometric algorithms for sensor networks. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 370:27–51, 01 2012.

[9] M.T. Goodrich and D. Strash. Succinct greedy geometric routing in the euclidean plane. In *Proceedings of the 20th International Symposium on Algorithms and Computation*, ISAAC '09, page 781–791. Springer-Verlag, 2009.

[10] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. Tarjan. Linear time algorithms for visibility and shortest path problems inside simple polygons. In *Proceedings of the Second Annual Symposium on Computational Geometry*, SCG '86, page 1–13. Association for Computing Machinery, 1986.

[11] J. Hershberger, V. Polishchuk, B. Speckmann, and T. Talvitie. Geometric kth shortest paths: The applet. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, SOCG'14, page 96–97. Association for Computing Machinery, 2014.

[12] C. Icking and R. Klein. Searching for the kernel of a polygon—a competitive strategy. In *Proceedings of the Eleventh Annual Symposium on Computational Geometry*, SCG '95, page 258–266. Association for Computing Machinery, 1995.

[13] C. Icking, R. Klein, and E. Langetepe. Self-approaching curves. *Mathematical Proceedings of the Cambridge Philosophical Society*, 125, 05 2002.

[14] D. G. Larman and P. McMullen. Arcs with increasing chords. *Mathematical Proceedings of the Cambridge Philosophical Society*, 72(2):205–207, 1972.

[15] T. Leighton and A. Moitra. Some results on greedy embeddings in metric spaces. *Discrete & Computational Geometry*, 44:686–705, 10 2008.

[16] J McCleary. *Geometry From a Differentiable Viewpoint.*, volume 2nd ed. Cambridge University Press, 2013.

[17] M. Nöllenburg, R. Prutkin, and I. Rutter. On self-approaching and increasing-chord drawings of 3-connected planar graphs. In Christian Duncan and Antonios Symvonis, editors, *Graph Drawing*, pages 476–487. Springer Berlin Heidelberg, 2014.

[18] C.H. Papadimitriou and D. Ratajczak. On a conjecture related to geometric routing. In Sotiris E. Nikoletseas and José D. P. Rolim, editors, *Algorithmic Aspects of Wireless Sensor Networks*, pages 9–17. Springer Berlin Heidelberg, 2004.

[19] H. Reisi Dehkordi, F. Frati, and J. Gudmundsson. Increasing-chord graphs on point sets. *Journal of Graph Algorithms and Applications*, 19, 08 2014.

[20] S. A. Robertson. Smooth curves of constant width and transnormality. *Bulletin of the London Mathematical Society*, 16(3):264–274, 1984.

[21] G. Rote. Curves with increasing chords. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 115, pages 1–12. Cambridge University Press, 1994.

[22] J. Wang and X. He. Succinct strictly convex greedy drawing of 3-connected plane graphs. *Theoretical Computer Science*, 532:80–90, 01 2014.

Appendix A

Finding the ratio between two distinct shortest self-approaching paths

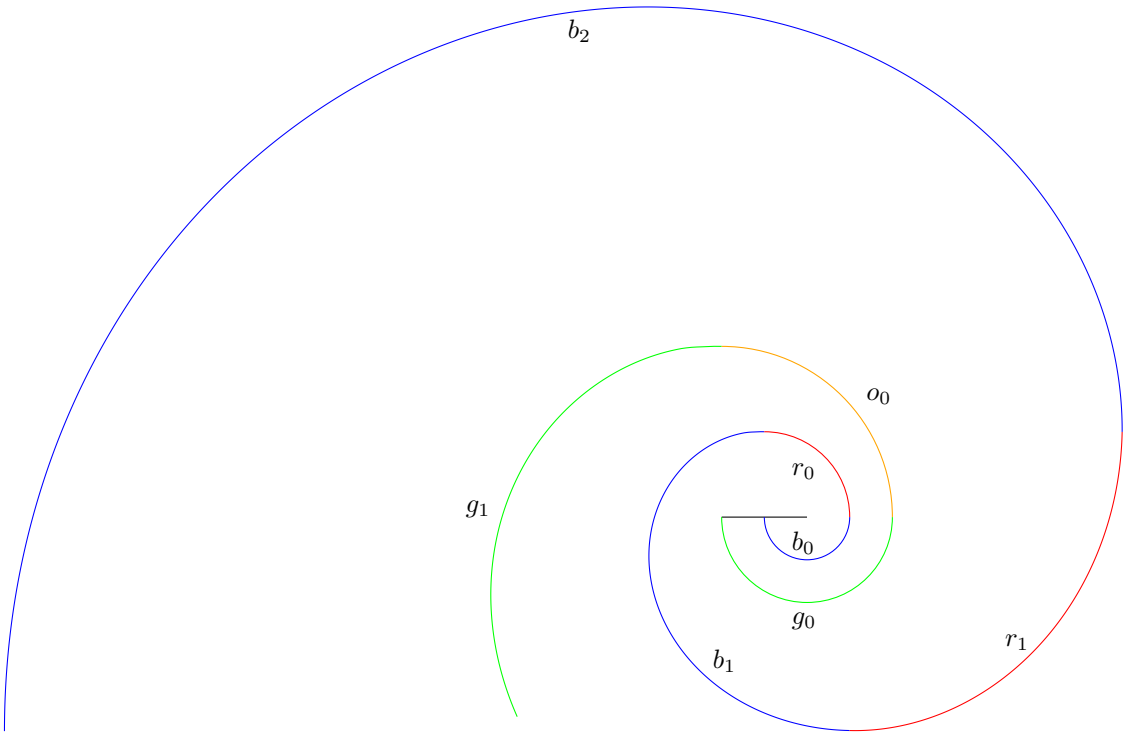


Figure A.1: Self-approaching paths γ (green & orange) and π (blue & red). For two sequential sections, the larger section is the involute of the smaller section, for example b_2 is an involute of b_1.

In this appendix we will describe how we calculate the ratio of the two self-approaching paths as described in Section 4.1.2 (Figure A.1). We will go over the Mathematica code that calculates the proper ratio. First we need some function definitions.

```
In[1]:= (* Assume that 0^0=1 *)
        p[0,0]=1;
        p[a_,b_]:=a^b;

        (* returns true iff point m lies to the left of a line going through points a and b *)
        isLeft[a_,b_,m_] := (b[[1]]-a[[1]]) (m[[2]]-a[[2]])-(b[[2]]-a[[2]]) (m[[1]]-a[[1]]) ≥ 0

        (* retuns true iff a line and a line segment intersect *)
        lineSegIntersect[lineA_,lineB_,segA_,segB_] := (isLeft[lineA,lineB,segA]
          && !isLeft[lineA,lineB,segB]) || (isLeft[lineB,lineA,segA] && !isLeft[lineB,lineA,segB]);

        (* returns true iff two line segments intersect *)
        SegSegInter[segA_,segB_,segC_,segD_] := lineSegIntersect[segA,segB,segC,segD]
          && lineSegIntersect[segC,segD,segA,segB];

        (* returns true iff a function f intersects with a line segment *)
        brokenSegFunIntersect[n_,segA_,segB_,ri_,ci_,offset_,start_,step_] :=
          SegSegInter[segA,segB,f[n,ri,start+(0 step)/8,ci,offset],f[n,ri,start+(1 step)/8,ci,offset]]
          || SegSegInter[segA,segB,f[n,ri,start+(1 step)/8,ci,offset],f[n,ri,start+(2 step)/8,ci,offset]]
          || SegSegInter[segA,segB,f[n,ri,start+(2 step)/8,ci,offset],f[n,ri,start+(3 step)/8,ci,offset]]
          || SegSegInter[segA,segB,f[n,ri,start+(3 step)/8,ci,offset],f[n,ri,start+(4 step)/8,ci,offset]]
          || SegSegInter[segA,segB,f[n,ri,start+(4 step)/8,ci,offset],f[n,ri,start+(5 step)/8,ci,offset]]
          || SegSegInter[segA,segB,f[n,ri,start+(5 step)/8,ci,offset],f[n,ri,start+(6 step)/8,ci,offset]]
          || SegSegInter[segA,segB,f[n,ri,start+(6 step)/8,ci,offset],f[n,ri,start+(7 step)/8,ci,offset]]
          || SegSegInter[segA,segB,f[n,ri,start+(7 step)/8,ci,offset],f[n,ri,start+(8 step)/8,ci,offset]];
```

Next we need to initialize values that define the involutes, first the ratios for the involute of order 0 is given for the different colored sections (Figure A.1). Thereafter, the constants for every colored section up to order 14 are provided, which were computed with enough signficance using Equation 2.3.

```
In[2]:= rb = 1;
        rr = 2;
        rg = 2;
        ro = 4;

        cb = {-5.141592653589793,  19.501172814903853,  -71.1046800598083, 259.0312364968812,
        -943.9509152267383, 3439.9357249349987, -12535.723161211947, 45682.34967342297,
        -166474.41610226588, 606661.6846328818, -2210780.538675379, 8056468.232744515,
        -29359169.420233447, 106989912.22261253};

        cr = {-17.707963267948966, 91.28516983579917, -377.8085380089388, 1433.1738201726052,
        -5278.356390637179, 19280.503350533123, -70292.80199770152, 256178.00286256624,
        -933567.153512947, 3402085.627172878, -12397792.94420039, 45179711.455358505,
        -164642715.27831963, 599986648.3472228};

        cg = {-10.283185307179586, 39.00234562980771, -142.2093601196166, 518.0624729937624,
        -1887.9018304534766, 6879.871449869997, -25071.446322423893, 91364.69934684594,
        -332948.83220453176, 1213323.3692657636, -4421561.077350758, 16112936.46548903,
        -58718338.840466894, 213979824.44522506, -779779642.5066359, 2841652442.9079223,
        -10355475016.412548, 37737149412.19475, -137520726330.87762, 501149410200.56854,
        -1826275486213.649, 6655265044031.606, -24252941651283.492, 88381931425565.06,
        -322079107550233.94, 1173712203921684.2};

        co = {-35.41592653589793, 182.57033967159833, -755.6170760178776, 2866.3476403452105,
        -10556.712781274358, 38561.006701066246, -140585.60399540304, 512356.0057251325,
        -1867134.307025894, 6804171.254345756, -24795585.88840078, 90359422.91071701,
        -329285430.55663925, 1199973296.6944456, -4372911094.556007, 15935647479.247122,
        -58072266984.50376, 211625426397.706, -771199807129.3015, 2810386032718.7905,
        -10241534787594.438, 37321931430194.5, -136007599892878.44, 495635314673340.94,
        -1806181164470387.2, 6582037845785234.0};
```

Thereafter, we give the actual functions for the involutes are given as per Equations 2.2 and 2.1.

```
In[3]:= a[r0_,i_,theta_,c_] := r0 p[theta,i]/i! + Sum[c[[j]] p[theta,i-j]/(i-j)!, {j,1,i}];

f[k_,r0_,theta_,c_,offset_] := {offset + Sum[(p[-1,i] a[r0,2 i,theta,c]) Cos[theta], {i,0,Floor[k/2]}]
  - Sum[(p[-1,i-1] a[r0,2 i-1,theta,c]) (-Sin[theta]), {i,1,Ceiling[k/2]}],
 Sum[(p[-1,i] a[r0,2 i,theta,c]) Sin[theta], {i,0,Floor[k/2]}]
  - Sum[(p[-1,i-1] a[r0,2 i-1,theta,c]) Cos[theta], {i,1,Ceiling[k/2]}]};
```

These functions compute the length of a section of an involute for a given order k and from a certain angle.

```
In[4]:= lb[k_,o_] := Integrate[Sqrt[f'[k,rb,theta,cb,0][[1]]^2+f'[k,rb,theta,cb,0][[2]]^2], {theta,o,π}];
        lr[k_,o_] := Integrate[Sqrt[f'[k,rr,theta,cr,-1][[1]]^2+f'[k,rr,theta,cr,-1][[2]]^2], {theta,o,2π}];
        lg[k_,o_] := Integrate[Sqrt[f'[k,rg,theta,cg,0][[1]]^2+f'[k,rg,theta,cg,0][[2]]^2], {theta,o,π}];
        lo[k_,o_] := Integrate[Sqrt[f'[k,ro,theta,co,-2][[1]]^2+f'[k,ro,theta,co,-2][[2]]^2], {theta,o,2π}];
```

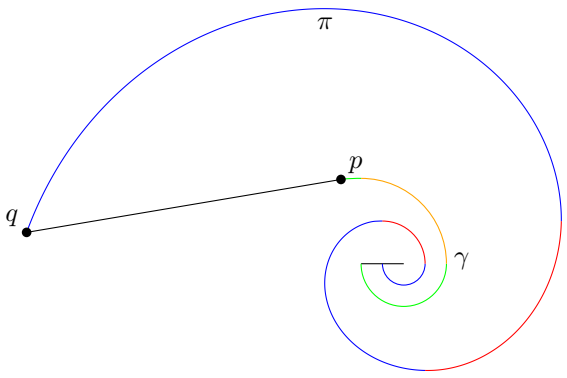


Figure A.2: Paths γ and π, such that segment $\overline{pq}$ is tangent to γ through p and π ends in q.

The following piece constructs self-approaching path γ up to a certain point p, calculates the tangent τ and construct π until it intersects with τ at point q. Then we can calculate the desired ratio by dividing $|\pi|$ by $|\gamma| + |pq|$ (Figure A.2).

```
In[5]:= (* we calculate for 8 orders of involutes *)
        For[k = 1, k < 14, k++,
          (* first we build the green section of order k *)
          For[i = 8, i ≥ 0, i--,
            (* derivative of the green section of order k *)
            fp[theta_] = D[f[k, rg, theta, cg, 0], theta];

            (* we construct up to: *)
```

```
  t = i/8*π/1;
  (* calculate the tangent at the point of construction *)
  tangentA = f[k, rg, t, cg, 0];
  tangentB = {tangentA[[1]] - 10000*fp[t][[1]], tangentA[[2]] - 10000*fp[t][[2]]};

  (* initialization of variables *)
  eps = 1;

  l = 3/2 π;
  r = 2 π;
  n = 1;
  ri = rr;
  ci = cr;
  offset = -1;

  (* find at which order of blue or red section the other path is intersected *)
  While[¬(brokenSegFunIntersect[n, tangentA, tangentB, rb, cb, 0, 0, π] ∨
    brokenSegFunIntersect[n, tangentA, tangentB, rr, cr, -1, 3/2 π, π/2]) ∧ n < 100, n++ ];

  (* determine if the other path is intersected in the red or blue section *)
  If[brokenSegFunIntersect[n, tangentA, tangentB, rr, cr, -1, 3/2 π, π/2], ,
    l = 0; r = π; ci = cb; ri = rb; offset = 0];

  (* using a binary search, find the point at which the other path is intersected *)
  While[eps > 0.001, If[SegSegIntersect[tangentA, tangentB, f[n, ri, l, ci, offset],
    f[n, ri, (l + r)/2, ci, offset]], r = (l + r)/2, l = (l + r)/2]; eps = r - l; ];
  thetaN = N[(l + r)/2];

  (* calculate the length of pi *)
  pi = ∑_{i=0}^{n-1} (lb[i, 0] + lr[i, (3/2)π]) + 1;

  If[ri == rb, pi = pi + lb[n, thetaN], pi = pi + lb[n, 0] + lr[n, thetaN]];

  (* calculate the length of gamma *)
  gamma = ∑_{i=0}^{k-1} (lg[i, 0] + lo[i, (3/2)π]) + lg[k,t]
    + ArcLength[Line[{f[k, rg, t, cg, 0], f[n, ri, thetaN, ci, offset]}]] + 4;

  (* print the output *)
  WriteString["stdout", ToString[k + (8-i)/12], " \t",ToString[N[pi / gamma]],"\n"];
];

(* now we build the orange section of order k *)
For[i = 4, i ≥ 1, i--,
  (* derivative of the orange section of order k *)
  fp[theta_] = D[f[k, ro, theta, co, -2], theta];

  (* we construct up to: *)
  t = 3/2 π + i/4*π/2;
  (* calculate the tangent at the point of construction *)
  tangentA = f[k, ro, t, co, -2];
  tangentB = {tangentA[[1]] - 10000*fp[t][[1]], tangentA[[2]] - 10000*fp[t][[2]]};

  (* initialization of variables *)
  eps = 1;

  l = 3/2 π;
  r = 2 π;
  n=1;
  ri = rr;
  ci = cr;
  offset = -1;
```

```
    (* find at which order of blue or red section the other path is intersected *)
    While[¬(brokenSegFunIntersect[n, tangentA, tangentB, rb, cb, 0, 0, π] ∨
      brokenSegFunIntersect[n, tangentA, tangentB, rr, cr, -1, 3/2 π, π/2]) ∧ n < 100, n++ ];

    (* determine if the other path is intersected in the red or blue section *)
    If[brokenSegFunIntersect[n, tangentA, tangentB, rr, cr, -1, 3/2 π, π/2], ,
      l = 0; r = π; ci = cb; ri = rb; offset = 0];

    (* using a binary search, find the point at which the other path is intersected *)
    While[eps > 0.001, If[SegSegIntersect[tangentA, tangentB, f[n, ri, l, ci, offset],
      f[n, ri, (l+r)/2, ci, offset]], r = (l + r)/2, l = (l + r)/2]; eps = r - l; ];
    thetaN = N[(l + r)/2];

    (* calculate the length of pi *)
    pi = ∑_{i=0}^{n-1} (lb[i, 0] + lr[i, (3/2)π]) + 1;

    If[ri == rb, pi = pi + lb [n, thetaN], pi = pi + lb[n, 0] + lr[n, thetaN]];

    (* calculate the length of gamma *)
    gamma = ∑_{i=0}^{k-1} (lg[i, 0] + lo[i, (3/2)π]) + lg[k,0] + lo[k, t]
      + ArcLength[Line[{f[k, ro, t, co, -2], f[n, ri, thetaN, ci, offset]}]] + 4;

    (* print the output *)
    WriteString["stdout", ToString[k + 8/12 + (4-i)/12], " \t",ToString[N[pi / gamma]],"\n"];
  ]
]
```

Finally, the output can be seen processed in Figure 4.7