

# Computation on Parametric Curves with an Application in Grasping

Yan-Bin Jia

Department of Computer Science  
Iowa State University  
Ames, IA 50011-1040, U. S. A.  
jia@cs.iastate.edu

## Abstract

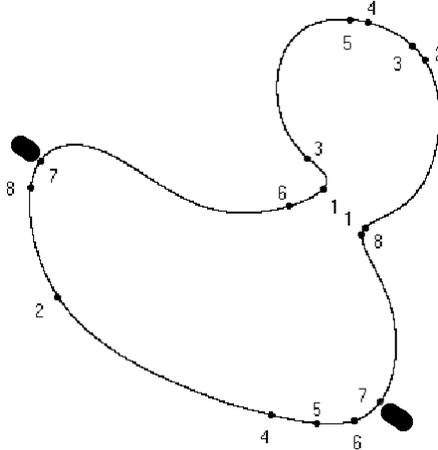
*Curved shapes are frequent subjects of maneuvers by the human hand. In robotics it is well known that antipodal grasps exist on curved objects and guarantee force closure under proper finger contact conditions. This paper presents an efficient algorithm that computes, up to numerical resolution, all pairs of antipodal points on a simple, closed, and twice continuously differentiable plane curve. Dissecting the curve into segments everywhere convex or everywhere concave, the algorithm marches simultaneously on a pair of such segments with provable convergence and interleaves marching with numerical bisection recursively. It makes use of new insights into the differential geometry at two antipodal points. We have avoided resorting to traditional nonlinear programming which would neither be quite as efficient nor guarantee to find all antipodal points. A byproduct of our result is a procedure that constructs all common tangent lines of two curves, achieving quadratic convergence rate. Dissection and the coupling of marching with bisection constitute an algorithm design scheme potentially applicable to computational problems involving curves and curved shapes.*

KEY WORDS—antipodal point, antipodal angle, inflection, monotonicity, common tangent, convergence rate, robot grasping

## 1 Introduction

Geometry often plays an essential role in robot tasks such as sensing, grasping, pushing, path planning, dexterous manipulation, and control. Sensing, manipulation, and planning strategies for polygonal and polyhedral objects have been studied extensively in the past. Many of these strategies are designed using combinatorial techniques that take advantage of the absence of local (differential) geometry.

Curved shapes, nevertheless, are very common in the real world. They share the differential nature with contact kinematics and dynamics which are responsible for dexterous maneuvers [2]. Processing of these shapes tends to be difficult because the nature of the problem is no longer combinatorial but nonlinear instead. The computational aspects have not been studied nearly to the same extent (as for polygons and polyhedra). In this paper we aim at introducing a scheme for curve computation within the context of one manipulation task — grasping.



**Figure 1:** Eight pairs of antipodal points on a curved shape. The points in each pair are numbered the same. Two fingers placed at any of these eight pairs of points will form a force-closure grasp of the shape in the presence of contact friction.

Grasping has remained one of the fundamental problems in robotics research. A grasp on an object is *force closure* if and only if arbitrary force and torque can be exerted on the object through the finger contacts. In the presence of friction, a pair of fingers placed on a two-dimensional (2D) object achieves force closure if and only if the line segment connecting the two points of contact lies within the contact friction cones at these points. Based on this result from planar mechanics, Nguyen [26] offered simple algorithms that synthesize independent grasp regions on polygons and polyhedra, with or without friction. Decomposing the configuration space into grasp rectangles, Ponce *et al.* [28] designed an algorithm that computes pairs of maximal-length segments on a piecewise smooth 2D shape where the placement of two fingers guarantees force closure with friction.

A force-closure grasp on a curved 2D object with friction is guaranteed if two fingers are placed at points whose inward normals are collinear and pointing at each other. Such a grasp is referred to as an *antipodal grasp* and the two points are referred to as antipodal points. Figure 1 shows eight pairs of antipodal points found on the boundary of a curved shape and an antipodal grasp achieved at one such pair. Implementing fine manipulation with “finger gaits”, Hong *et al.* [12] proved the existence of at least two pairs of antipodal points on a smooth convex 2D object. Chen and Burdick [7] gave the first algorithm for computing antipodal points on 2D and 3D shapes: their idea was to convert the problem into one of finding the critical points of a grasping energy function so an existing constrained optimization method could be applied. The early mentioned work by Ponce *et al.* [28], however, does not guarantee to return segments containing antipodal points.

The reduction of antipodal point computation to nonlinear programming has several drawbacks. First, such an approach is incomplete in the sense that it does not guarantee to find any antipodal points, not to mention all of them. Second, it is inefficient as many initial guesses are often required to find an antipodal pair. Third, the performance tends to heavily depend on the quality of the heuristics used.

This paper presents an efficient algorithm that finds all antipodal grasps on a 2D curved shape up to numerical resolution. The algorithm employs a fast procedure that constructs common tangents

of two curve segments. The novelty of the algorithm lies in (a) its dissection of the curve into segments turning monotonically in one direction; and based on the dissection, (b) its combination of “marching” and numerical bisection with provable convergence. The correctness of the algorithm will be ensured by the local geometry at antipodal points. The completeness<sup>1</sup> in finding all antipodal points (up to numerical resolution) will be established by performing curvature-based analyses.

## 1.1 Other Related Work

Mishra *et al.* [23] gave upper bounds on the numbers of frictionless fingers that are sufficient for equilibrium and force-closure grasps, respectively, on objects with piecewise smooth boundaries. They also provided linear-time algorithms that synthesize such grasps for polygonal and polyhedral objects. Tighter bounds were later obtained by Markenscoff *et al.* [20] on immobilizing 2D and 3D objects with piecewise smooth boundaries. Force closure testing was formulated by Trinkle [43] as a linear program whose objective function measures the robustness of a grasp. In [29], Ponce *et al.* reduced the problem of computing force-closure grasps of a polyhedral object to that of projecting a polytope onto some linear space. Blake and Taylor [4] gave a geometric classification of two-fingered frictional grasps of smooth contours that include antipodal grasps. Using the parallel-jaw gripper to orient parts bounded by algebraic curves, Rao and Goldberg [35] offered an efficient algorithm that produces the shortest plan based on a “grasp” function.

Li and Sastry [18] proposed three quality measures of grasps by multifingered robotic hands. Algorithms for finding optimal grasps of polygons were developed by a number of researchers including Markenscoff and Papadimitriou [21] and Mirtich and Canny [22]. Mishra [25] compared various grasp metrics from the viewpoint of finger and computational complexities. He [24] also gave the earliest results on the existence of modular fixture designs.<sup>2</sup> The problem of optimal fixture design was later studied by Brost and Goldberg [5] and by other researchers. Teichmann and Mishra [42] showed that hands controlled by reactive algorithms can grasp polygonal objects when equipped with light-beam sensors, or objects with smooth boundary when equipped with simple distance and angle sensors. The designs of these hands are detailed in Teichmann’s Ph.D. thesis [41]. We refer the reader to a survey [3] conducted by Bicchi and Kumar on grasping and contact.

Two vertices of a polygon are antipodal if they are incident on parallel supporting lines. The polygon’s diameter is the distance between the two furthest vertices (which must be antipodal). Preparata and Shamos [32] described a linear-time algorithm that finds the diameter of a convex polygon with  $n$  vertices by enumerating all  $O(n)$  pairs of antipodal vertices in a traversal. This algorithm was extended by them to determine the diameter of a set of  $n$  points in the plane in optimal  $O(n \log n)$  time.

For  $n$  points in three dimensions, the number of antipodal pairs can be  $O(n^2)$  while the diameter can be achieved by  $2n - 2$  pairs of points, as shown by Grunbaum [11]. Clarkson and Shor [8] presented a randomized algorithm that computes the diameter in  $O(n \log n)$  expected time. Near-linear-time deterministic algorithms were developed by Chazelle *et al.* [6] and by Matoušek and Schwarzkopf [38] through derandomizing Clarkson and Shor’s algorithm while the fastest running time  $O(n \log^2 n)$  was achieved by Bepamyatnikh [1] and by Ramos [34]. Erdős [9] showed that  $O(n^2)$  pairs of points can attain the diameter in dimensions  $d > 3$ . Yao [44] offered a different

---

<sup>1</sup>The algorithm does not find all force-closure grasps in the presence of friction. In this sense it is incomplete. However, antipodal grasps are often more robust to uncertainties in finger placement than other force-closure grasps.

<sup>2</sup>These results were for rectilinear parts.

approach which led to an  $o(n^2)$ -time algorithm for computing the diameter of a point set in  $d$  dimensions.

Preparata and Hong [31] developed a “walking” strategy to construct common supporting lines of two convex polygons in linear time. Their method alternatively marches over the vertices of the two polygons such that the next vertex to reach on one polygon is always the point of tangency of the supporting line from the current vertex on the other polygon. The procedure `Common-Tangent` in Section 2.3 for computing common tangents of two curve segments can be viewed as the continuous counterpart of the “walking” strategy except tangent lines (instead of supporting lines) are used to determine the next steps.

In the preprocessing we will dissect a curve at its points of inflection. In [10], Goodman gave an upper bound on the number of inflection points on parametric spline curves. Sakai [37] obtained the distribution of inflection points and cusps on a parametric rational cubic curve. Manocha and Canny [19] presented a method to detect cusps on rational curves of arbitrary degree and described the use of the Sturm sequence in finding inflection points.

## 1.2 Notation and Overview

Throughout the paper we will consider a curve  $\alpha(u)$  that is regular<sup>3</sup>, simple, closed, and twice continuously differentiable. A counterclockwise movement along the curve increases the value of the parameter  $u$ . In case no ambiguity arises, the parameter  $u$  also refers to the point  $\alpha(u)$  on the curve. Denote by  $T(u) = \alpha'(u)/\|\alpha'(u)\|$  the unit tangent of  $\alpha$  at  $u$  and by  $N(u)$  its unit inward normal such that  $T(u) \times N(u) = 1$ .

Without loss of generality, the *curvature*<sup>4</sup>  $\kappa(u)$  of  $\alpha$  can be zero at only a finite number of points. The *total curvature* of a segment of  $\alpha$  over  $[a, b]$  is given by  $\Phi(a, b) = \int_a^b \kappa(u) \|\alpha'(u)\| du$ .<sup>5</sup> This integral measures the amount of tangent rotation as a point moves from  $a$  to  $b$  along the curve.

For clarity of presentation, we will focus on the case that  $\alpha$  is *unit-speed*, that is,  $\|\alpha'\| = 1$  everywhere. Thus  $\alpha$  is parameterized with arc length. Section 2 will present an algorithm that constructs all common tangents of two curve segments under several conditions that can be established by the preprocessing described in Section 5. Section 3 will introduce a classification of antipodal points based on differential geometry. This classification will lead to the design of an algorithm in Section 4 that finds all antipodal points. The running time of the algorithm will be analyzed in Section 6.

Section 7 will discuss how the results can be easily extended to arbitrary-speed curves for which some experiments will be presented in Section 8.

## 2 Common Tangent

In this section we will introduce an algorithm that constructs common tangents of two segments on the curve  $\alpha$  under some conditions to be introduced shortly. In the end of the section we will explain how the algorithm can be extended to find all common tangents of two curves.

---

<sup>3</sup>That is,  $\|\alpha'(u)\| \neq 0$  for any  $u$  in the curve domain.

<sup>4</sup>Two equivalent definitions of curvature are given in Appendix C.

<sup>5</sup>The total curvature has a closed form if it is within  $(0, 2\pi]$  and can almost always be fast computed otherwise. More on total curvature is discussed in Appendix C.

Two segments  $\mathcal{S}$  and  $\mathcal{T}$  of  $\alpha$  are defined on subdomains  $(s_a, s_b)$  and  $(t_a, t_b)$ , respectively, where  $(s_a, s_b) \cap (t_a, t_b) = \emptyset$ . Here  $s_a < s_b$  always holds but  $t_a > t_b$  is allowed, in which case  $(t_a, t_b)$  refers to the interval  $(t_b, t_a)$ . The following conditions are satisfied by  $\mathcal{S}$  and  $\mathcal{T}$ :

- (i)  $\kappa > 0$  everywhere or  $\kappa < 0$  everywhere on  $\mathcal{S}$  and  $\mathcal{T}$ , with  $\kappa = 0$  possible only at  $s_a, s_b, t_a, t_b$ .
- (ii)  $N(s_a) + N(t_a) = 0$  and  $N(s_b) + N(t_b) = 0$ .
- (iii)  $-\pi \leq \Phi(s_a, s_b) = -\Phi(t_a, t_b) \leq \pi$ .

Section 5 will describe how to preprocess  $\alpha$  to generate pairs of such segments.

Condition (i) states that the inward normal  $N$  rotates in one direction when either  $\mathcal{S}$  or  $\mathcal{T}$  is traversed. We call such a curve segment *monotone*. Under conditions (i)–(iii), every point  $s$  on  $\mathcal{S}$  uniquely corresponds to a point  $t$  on  $\mathcal{T}$  with the opposite normal. More precisely, the equation

$$N(s) + N(t) = 0, \quad \text{or equivalently,} \quad (1)$$

$$T(s) + T(t) = 0. \quad (2)$$

defines a one-to-one correspondence between the points on  $\mathcal{S}$  and the points on  $\mathcal{T}$ . Consider the equation  $\eta(s, t) = N(s) \times N(t) = 0$ . The Frenet formulas for unit-speed plane curves state that  $T'(t) = \kappa(t)N(t)$  and  $N'(t) = -\kappa(t)T(t)$ . Hence  $\frac{\partial \eta}{\partial t} = N(s) \times (-\kappa(t)T(t)) = -\kappa(t) \neq 0$  under (1). By the Implicit Function Theorem, the equation  $\eta(s, t) = 0$  defines  $t$  as a function of  $s$  on  $(s_a, s_b)$ . From now on, we will refer to  $t$  as the *opposite point* of  $s$ .

Differentiate (2) with respect to  $s$  and then substitute (1) in:

$$\begin{aligned} 0 &= \kappa(s)N(s) + \kappa(t)\frac{dt}{ds}N(t) \\ &= \left( \kappa(s) - \kappa(t)\frac{dt}{ds} \right) N(s). \end{aligned}$$

Thus  $\kappa(s) - \kappa(t)\frac{dt}{ds} = 0$  and

$$\frac{dt}{ds} = \frac{\kappa(s)}{\kappa(t)}. \quad (3)$$

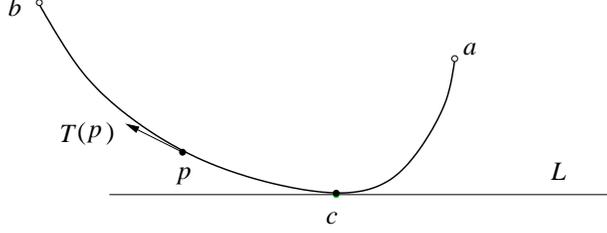
## 2.1 Number of Common Tangents

Following conditions (i)–(iii), no two points, one on the segment  $\mathcal{S}$  and the other on the segment  $\mathcal{T}$ , can have the same inward normal. Thus any common tangent of  $\mathcal{S}$  and  $\mathcal{T}$  must be in contact with  $\alpha$  at points with opposite inward normals.

**Lemma 1** *Any tangent line of a monotone curve segment over  $(a, b)$  with total curvature  $\Phi(a, b) \in [-\pi, \pi]$  intersects the segment exactly once.*

**Proof** Without loss of generality, suppose a tangent line  $L$  is horizontal and the segment is locally above the point of tangency  $c$  (see Figure 2). Consider a point  $p$  moving on the segment from  $c$  to  $b$ . Because the segment is monotone,  $|\Phi(c, b)| < |\Phi(a, b)| \leq \pi$  and  $p$  always moves upward. Therefore  $L$  cannot intersect the segment over  $(c, b)$ . Similarly,  $L$  cannot intersect the segment over  $(a, c)$ . Thus  $L$  intersects the curve segment over  $(a, b)$  exactly once.  $\square$

Using the above lemma, we can now bound the number of common tangents.

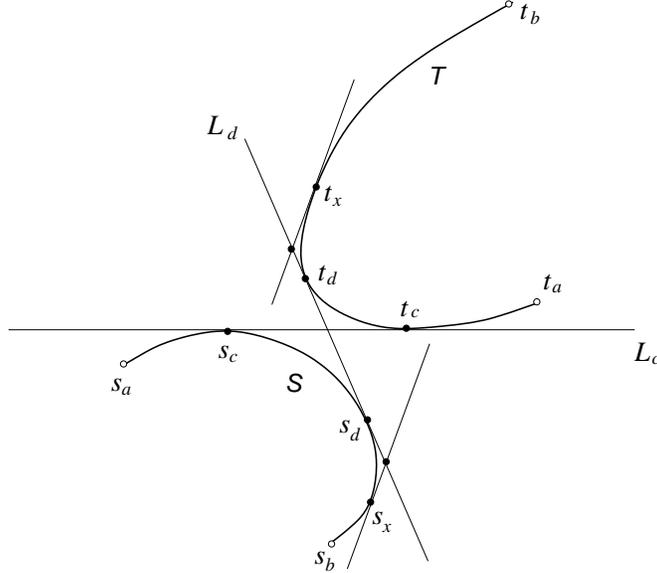


**Figure 2:** A monotone curve segment with total curvature in  $[-\pi, \pi]$  lies on one side of every tangent line.

**Theorem 2** Let  $\mathcal{S}$  and  $\mathcal{T}$  be two segments of  $\alpha$  defined over  $(s_a, s_b)$  and  $(t_a, t_b)$ , respectively, and satisfying conditions (i)–(iii). They have at most two common tangents.

**Proof** We need only consider the case of two common tangents  $L_c$  and  $L_d$  of  $\mathcal{S}$  and  $\mathcal{T}$  and show that a third one cannot exist. Let  $s_c$  and  $t_c$  be the points of tangency of  $L_c$ ; and let  $s_d$  and  $t_d$  be the points of tangency of  $L_d$ . Suppose  $s_c < s_d$ . Following Lemma 1,  $L_c$  intersects neither  $\mathcal{S}$  nor  $\mathcal{T}$  more than once. For the ease of argument (but without loss of generality), we assume that  $L_c$  is horizontal and  $s_c$  is to the left of  $t_c$  on  $L_c$ . Two cases arise depending on the positions of  $\mathcal{S}$  and  $\mathcal{T}$  relative to  $L_c$ .

**Case 1:**  $\mathcal{S}$  and  $\mathcal{T}$  on opposite sides of  $L_c$ . Since  $\mathcal{S}$  cannot intersect  $L_c$  or  $L_d$  more than once, it is bounded by  $L_c$  and  $L_d$ . Similarly,  $\mathcal{T}$  is bounded by  $L_c$  and  $L_d$  on the opposite side. This configuration is shown in Figure 3. It is clear that no common tangent can be incident on any point in  $(s_c, s_d)$ , or equivalently, on any point in  $(t_c, t_d)$ . Consider an arbitrary tangent line

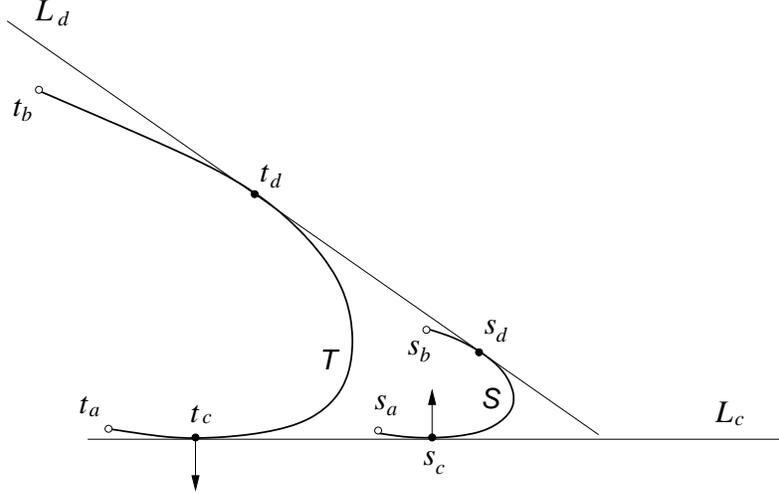


**Figure 3:** Two curve segments on opposite sides of every common tangent.

of  $\mathcal{S}$  incident on some point  $s_x \in (s_d, s_b)$  and the tangent line of  $\mathcal{T}$  incident on its opposite point  $t_x \in (t_d, t_b)$ , where  $N(s_x) + N(t_x) = 0$ . These two tangent lines always intersect  $L_d$  on different sides of the line segment  $\overline{s_d t_d}$ . Therefore they will never coincide, implying that no

common tangent can be incident on a point in  $(s_d, s_b)$ . Similarly, no common tangent can be incident on a point in  $(s_a, s_c)$ .

**Case 2:**  $\mathcal{S}$  and  $\mathcal{T}$  on the same side of  $L_c$ . Without loss of generality, suppose  $t_c$  is to the left of  $s_c$  on  $L_c$ , as shown in Figure 4. Then  $L_d$  must have a negative slope under the condition that  $|\Phi(s_a, s_b)| \leq \pi$ . Clearly, any line through a point in  $(s_c, s_d)$  and its opposite point in  $(t_d, t_c)$



**Figure 4:** Two curve segments on the same side of every common tangent.

cannot be tangent to either  $\mathcal{S}$  or  $\mathcal{T}$ . Meanwhile, any tangent line of  $\mathcal{T}$  incident on a point in  $(t_d, t_b)$  cannot intersect  $\mathcal{S}$ ; let alone be a common tangent. Similarly, any tangent line of  $\mathcal{T}$  incident on a point  $(t_a, t_c)$  will not intersect  $\mathcal{S}$ ; let alone be a common tangent line.

□

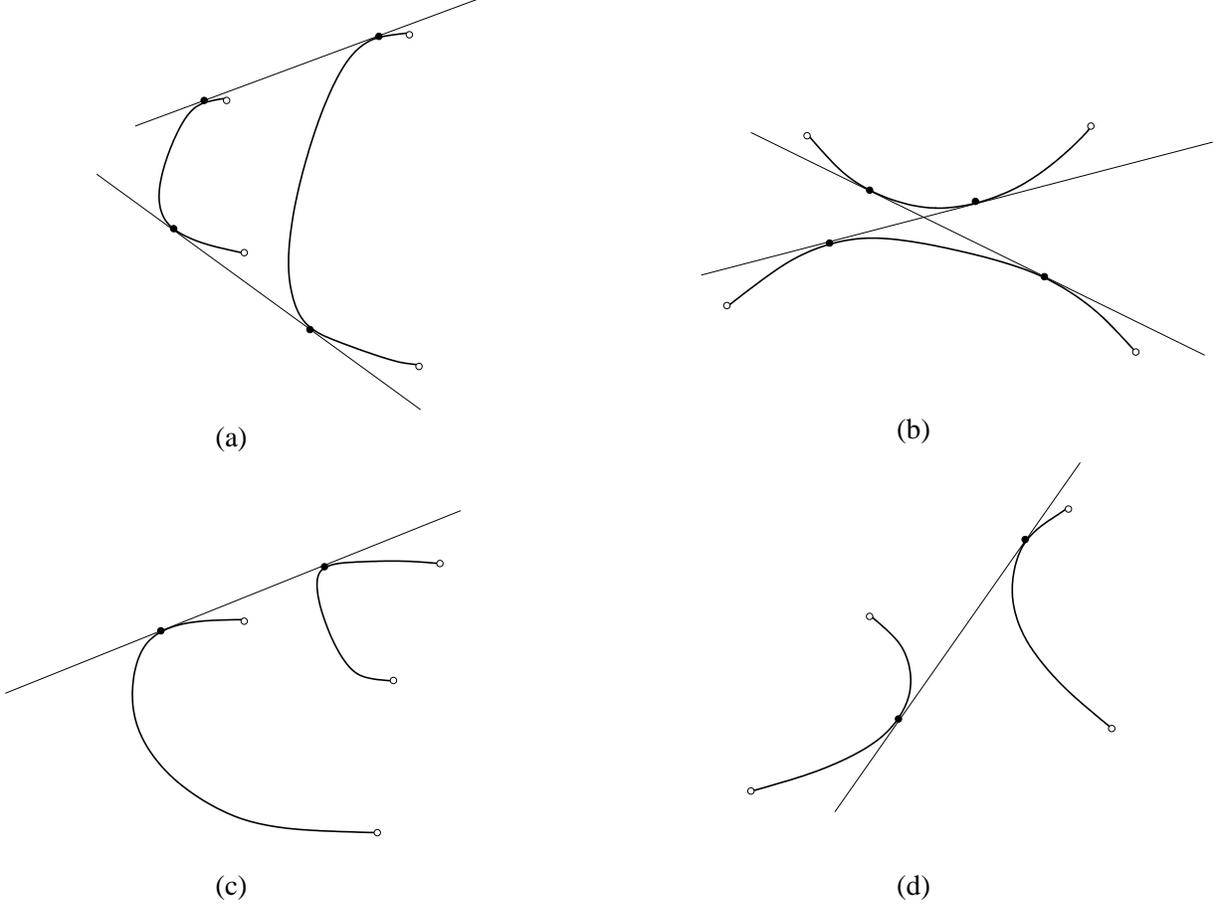
## 2.2 Configurations of Common Tangents

The proof of Theorem 2 in the previous section has also established the following fact:

**Corollary 3** *Let  $\mathcal{S}$  and  $\mathcal{T}$  be a pair of segments that satisfy conditions (i)–(iii). Then they are always on the same side of every common tangent or always on different sides of every common tangent.*

Following the corollary there are a total of four configurations (shown in Figure 5) in which at least one common tangent of  $\mathcal{S}$  and  $\mathcal{T}$  exists:

- (a) Two common tangents with the curve segments on the same side.
- (b) Two common tangents with the segments on different sides.
- (c) One common tangent with the segments on the same side.
- (d) One common tangent with the segments on different sides.



**Figure 5:** Four configurations where  $\mathcal{S}$  and  $\mathcal{T}$  share at least one common tangent.

In configurations (a) and (c) the two curve segments bend in the same direction while in (b) and (d) they bend in opposite directions. The bending direction of the curve  $\alpha$  at a point  $u$ , where  $\kappa(u) \neq 0$ , is indicated by  $\kappa(u)N(u)$ . Let  $s_m = \frac{s_a + s_b}{2}$  and  $t_m$  be its opposite point. We check the dot product

$$\kappa(s_m)N(s_m) \cdot \kappa(t_m)N(t_m) = -\kappa(s_m)\kappa(t_m).$$

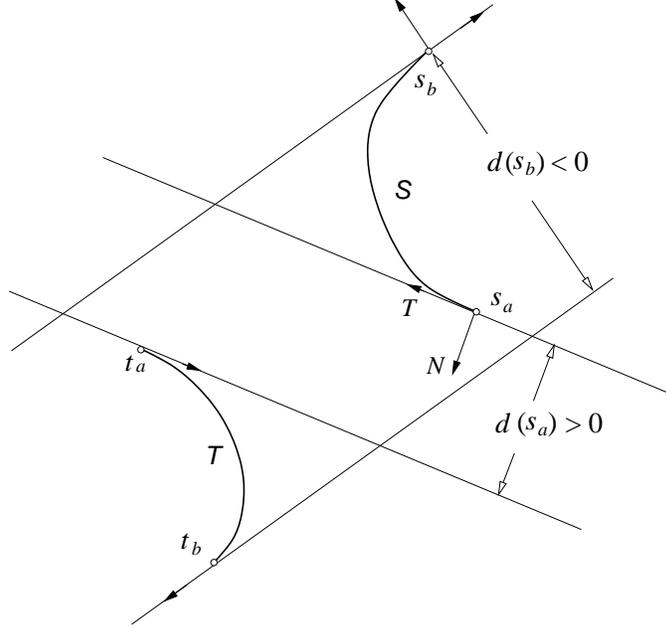
If the dot product is positive, then the two segments  $\mathcal{S}$  and  $\mathcal{T}$  bend in the same direction. If it is negative, the two segments bend in opposite directions.

The common tangent algorithm will rely on the above classification of configurations. Below we look at how to recognize each configuration as well as the configuration where no common tangent exists.

Define the function  $d(s)$  as the translational distance along the normal  $N(s)$  from the tangent line of  $\mathcal{S}$  at  $s$  to the tangent line of  $\mathcal{T}$  at its opposite point  $t$ . Thus we have

$$d(s) = N(s) \cdot (\alpha(t) - \alpha(s)).$$

In the example shown in Figure 6,  $d(s_a) > 0$  and  $d(s_b) < 0$ . The function  $d$  is continuous since the curve  $\alpha$  is continuously differentiable. Obviously,  $d(s) = 0$  if and only if the tangent line through



**Figure 6:** The values of the function  $d$  at  $s_a$  and  $s_b$  have opposite signs.

$s$  is a common tangent. It follows from the monotonicity condition (i) that  $d'(s) \neq 0$  whenever  $d(s) = 0$ .

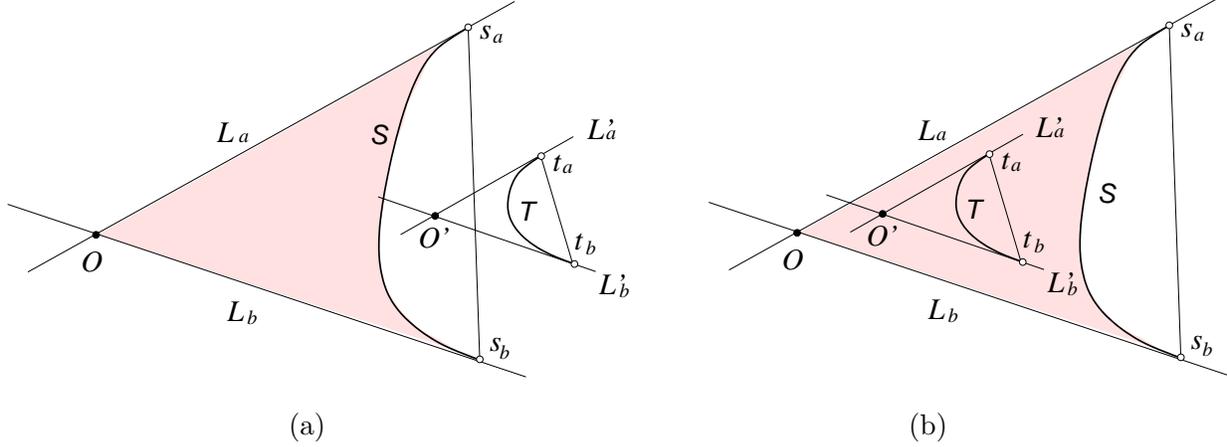
When  $d(s_a)$  and  $d(s_b)$  have different signs, there exists some  $s_c \in (s_a, s_b)$  such that  $d(s_c) = 0$ . Namely, the line passing through  $s_c$  and its opposite point on  $\mathcal{T}$  is a common tangent. Because there are at most two common tangents and the function  $d$  has only simple zeros, we can infer that this is the unique common tangent of  $\mathcal{S}$  and  $\mathcal{T}$  and the configuration is either (c) or (d) in Figure 5. These two configurations can be further distinguished by comparing the bending directions of  $\mathcal{S}$  and  $\mathcal{T}$ .

When  $d(s_a)$  and  $d(s_b)$  have the same sign,  $\mathcal{S}$  and  $\mathcal{T}$  may have two common tangents as in configuration (a) or (b) or they may not have a common tangent at all. Let  $L_a$  and  $L_b$  be the tangent lines of  $\mathcal{S}$  at  $s_a$  and  $s_b$ , respectively, and  $O$  be their intersection point.<sup>6</sup> Let  $L'_a$  and  $L'_b$  be the tangent lines of  $\mathcal{T}$  at  $t_a$  and  $t_b$ , respectively, and  $O'$  be their intersection. Since  $N(t_a) + N(s_a) = 0$  and  $N(t_b) + N(s_b) = 0$ , we know that  $L_a \parallel L'_a$  and  $L_b \parallel L'_b$ .

How to distinguish configuration (a) from a configuration where  $\mathcal{S}$  and  $\mathcal{T}$  bend in the same direction but do not have a common tangent? Since the total curvatures  $\Phi(s_a, s_b) = -\Phi(t_a, t_b) \in [-\pi, \pi]$ ,  $\mathcal{S}$  lies inside  $\triangle s_a O s_b$  and  $\mathcal{T}$  lies inside  $\triangle t_a O' t_b$ .

- If none of these two triangles contains the other, then  $\mathcal{S}$  and  $\mathcal{T}$  do not have a common tangent under the condition that  $d(s_a)$  and  $d(s_b)$  have the same sign. This is shown in Figure 7(a).
- Otherwise, suppose that  $\triangle t_a O' t_b$  is inside  $\triangle s_a O s_b$ . To share two common tangents,  $\mathcal{T}$  must be in the shaded region in Figure 7(b). Since  $\mathcal{S}$  and  $\mathcal{T}$  bend in the same direction, we see that  $\mathcal{T}$  is in the shaded region whenever  $t_a$  and  $t_b$  are. In fact, we need only check if  $t_a$  lies in

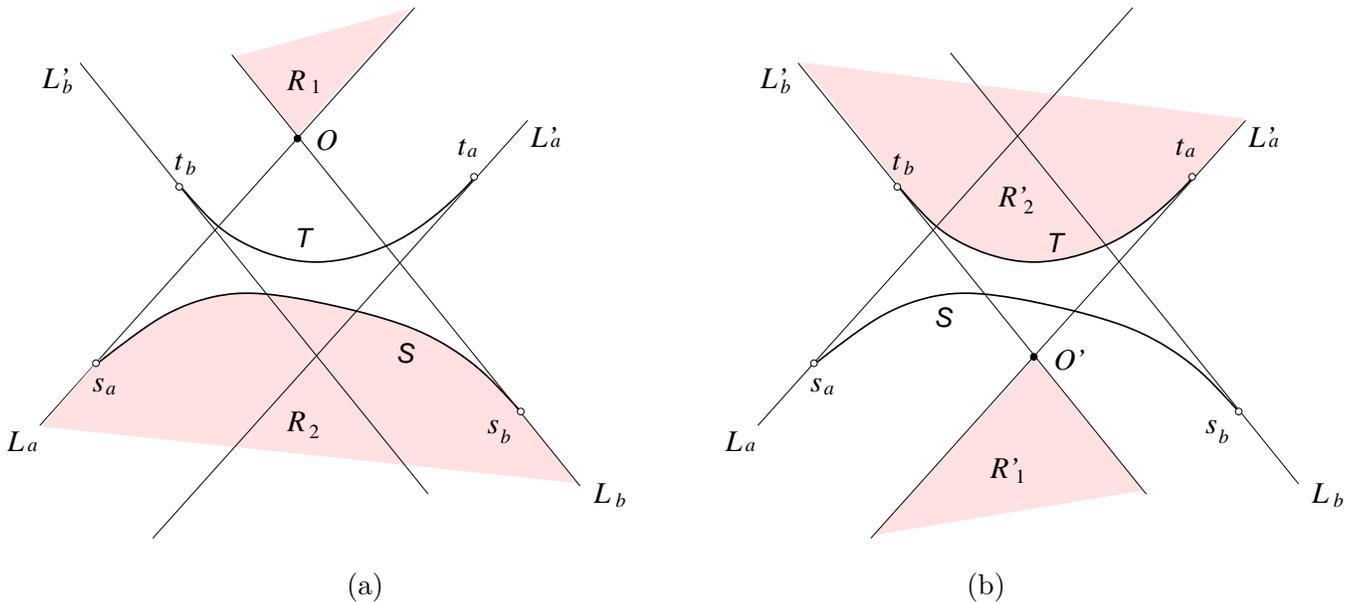
<sup>6</sup>Such an intersection does not exist if  $|\Phi(s_a, s_b)| = \pi$ . In this case, the triangle  $\triangle s_a O s_b$  becomes an open region bounded by  $L_a$ ,  $L_b$  and the line segment  $s_a s_b$ .



**Figure 7:** Two cases of  $\mathcal{S}$  and  $\mathcal{T}$  bending in the same direction: (a) no common tangent; and (b) two common tangents. In (a), neither of  $\triangle s_a O s_b$  and  $\triangle t_a O' t_b$  contains the other. In (b),  $\triangle s_a O s_b$  contains  $\triangle t_a O' t_b$ .

the shaded region. This is because under the condition that  $d(s_a)$  and  $d(s_b)$  have the same sign,  $t_b$  is in the shaded region if and only if  $t_a$  is. A procedure for testing if a point lies in the shaded region is described in Appendix B.

How to distinguish configuration (b) in Figure 5 from a configuration where  $\mathcal{S}$  and  $\mathcal{T}$  bend in the opposite directions but do not have a common tangent? The two tangent lines  $L_a$  and  $L_b$  partition the plane into four regions. As shown in Figure 8(a), we let  $R_1$  be the region that neither contains nor borders  $\mathcal{S}$ , and let  $R_2$  be the open region bounded by  $\mathcal{S}$ ,  $L_a$ , and  $L_b$ . Similarly, as shown in



**Figure 8:** The curve segments  $\mathcal{S}$  and  $\mathcal{T}$  bend in opposite directions and have two common tangents.

Figure 8(b), we let  $R'_1$  be the region formed by  $L'_a$  and  $L'_b$  that neither contains nor borders  $\mathcal{T}$  and



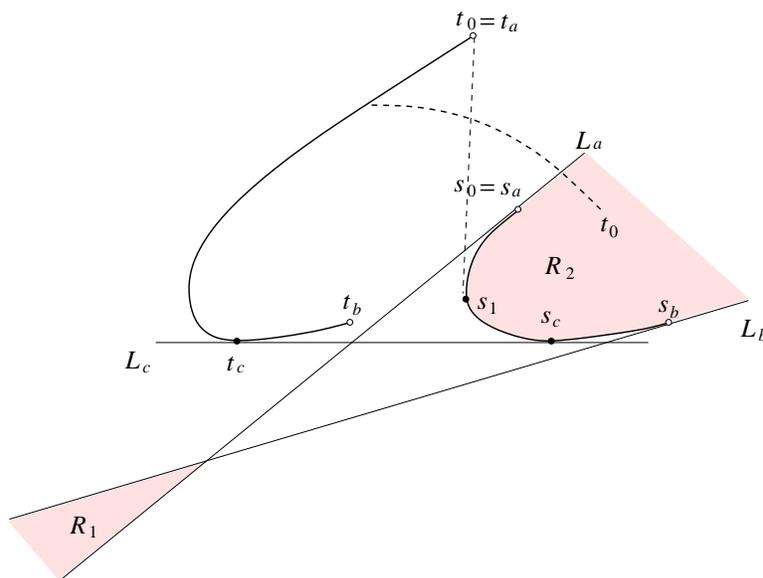
$$N(s_{i+1}) + N(t_{i+1}) = 0. \quad (5)$$

The point  $s_{i+1} \in (s_i, s_b)$  is found by a primitive **point-of-tangency** described in Appendix A. Such a point of tangency always exists under assumptions 1 and 2. The procedure call for finding the two points of tangency  $s_c$  and  $t_c$  is referred to as **Common-Tangent**( $s_a, s_b, t_a, t_b$ ).

**Lemma 4** *The two sequences  $\{s_i\}$  and  $\{t_i\}$ , starting at  $s_0 = s_a$  and  $t_0 = t_a$  and generated according to (4) and (5), never move past the two points of tangency  $s_c$  and  $t_c$ . Furthermore, the arc lengths  $|s_i - s_c| > |s_{i+1} - s_c|$  and  $|t_i - t_c| > |t_{i+1} - t_c|$  for all  $i \geq 0$ .*

**Proof** It suffices to show that the sequence  $\{s_i\}$  will not pass  $s_c$ . That  $\{t_i\}$  does not pass  $t_c$  will then follow from the one-to-one correspondences between  $s_i$  and  $t_i$  for all  $i$  and between  $s_c$  and  $t_c$ . Here we only give the proof for configuration (a) in Figure 9 in which  $\mathcal{S}$  and  $\mathcal{T}$  are on the same side of the tangent line  $L_a$ . The proof for configuration (b) is similar.

As shown in Figure 10, the segment  $\mathcal{S}$  and its two tangent lines  $L_a$  and  $L_b$  at  $s_a$  and  $s_b$ , respectively, divide the plane into three regions:  $R_1$ ,  $R_2$ , and the rest of the plane. Without loss of



**Figure 10:** Proof of Lemma 4.

generality, the common tangent  $L_c$  is drawn horizontal. The two regions  $R_1$  and  $R_2$  (excluding  $\mathcal{S}$ ) contain all the points not lying on any tangent line of  $\mathcal{S}$ .<sup>8</sup>

We now prove that the endpoint  $t_0 = t_a$  lies in neither  $R_1$  nor  $R_2$ .

- Since  $|\Phi(t_a, t_b)| \leq \pi$ , we have  $|\Phi(t_a, t_c)| < \pi$ . Thus  $t_0 = t_a$  must be above the tangent line  $L_c$ . Similarly,  $|\Phi(s_a, s_c)| < \pi$  and  $|\Phi(s_c, s_b)| < \pi$ . The intersection of  $L_a$  and  $L_b$  must be below the horizontal line  $L_c$ ,<sup>9</sup> and so must be the region  $R_1$ . Therefore  $t_0 \notin R_1$ .

<sup>8</sup>The region  $R_1$  will disappear if the total curvature of  $\mathcal{S}$  is  $\pi$  or  $-\pi$ .

<sup>9</sup>The intersection does not exist when  $\Phi(s_a, s_b) = \pi$ .

- Assume  $t_0 \in R_2$ . Then the section of  $\mathcal{T}$  over  $(t_a, t_c]$  would have to intersect  $L_a$  to enter  $R_2$ . This, however, would imply that  $|\Phi(t_a, t_c)| > |\Phi(s_a, s_c)|$ , contradicting the condition that they must be equal.

Therefore  $t_0 \notin R_1$  and  $t_0 \notin R_2$ . So  $t_0$  must be on some tangent line of  $\mathcal{S}$ . Let  $s_1$  be the point of tangency on  $\mathcal{S}$ . Because  $t_0$  is above  $L_c$ , it follows that  $s_1 \notin (s_c, s_b)$ . Hence  $s_1 \in (s_a, s_c)$  and  $|s_0 - s_c| > |s_1 - s_c|$ . Generalizing the above argument to the curve sections over  $(s_i, s_b)$  and  $(t_i, t_b)$ , we conclude that  $s_{i+1} \in (s_i, s_c)$  and  $|s_i - s_c| > |s_{i+1} - s_c|$  for all  $i$ .  $\square$

Lemma 4 establishes that the sequences  $\{s_i\}$  and  $\{t_i\}$  are monotone and bounded by  $s_c$  and  $t_c$ . So they must converge to, say,  $s^*$  and  $t^*$  such that  $(\alpha(s^*) - \alpha(t^*)) \times T(s^*) = 0$ . Hence  $s^* = s_c$  and  $t^* = t_c$ .

Next, we determine the order of convergence of  $\{s_i\}$  and  $\{t_i\}$ . Let  $h$  be the iteration function implicitly defined by (4) and (5) such that  $s_{i+1} = h(s_i)$ . Apply Taylor's expansion on  $h$  at  $s^*$  and use the fact that  $s^* = h(s^*)$ :

$$\begin{aligned} s_{i+1} - s^* &= h(s_i) - h(s^*) \\ &= h'(s^*)(s_i - s^*) + h''(s^*)(s_i - s^*)^2 + \dots \end{aligned}$$

The local convergence rate [40, p. 264] is determined by the least integer  $k > 0$  such that  $h^{(k)}(s^*) \neq 0$ . When  $k = 1$ , the local convergence rate is linear only if  $|h'(s^*)| < 1$ ,

Differentiate equation (4) with respect to  $s_i$ :

$$\left( T(s_{i+1}) \frac{ds_{i+1}}{ds_i} - T(t_i) \frac{dt_i}{ds_i} \right) \times T(s_{i+1}) + (\alpha(s_{i+1}) - \alpha(t_i)) \times N(s_{i+1}) \kappa(s_{i+1}) \frac{ds_{i+1}}{ds_i} = 0.$$

Simplify the above equation and substitute (3) in:

$$-T(t_i) \times T(s_{i+1}) \frac{\kappa(s_i)}{\kappa(t_i)} + \|\alpha(s_{i+1}) - \alpha(t_i)\| \kappa(s_{i+1}) \frac{ds_{i+1}}{ds_i} = 0.$$

So we obtain the derivative of the iteration function:

$$\begin{aligned} h'(s_i) &= \frac{ds_{i+1}}{ds_i} \\ &= \frac{T(t_i) \times T(s_{i+1})}{\|\alpha(s_{i+1}) - \alpha(t_i)\| \kappa(s_{i+1})} \frac{\kappa(s_i)}{\kappa(t_i)}. \end{aligned} \tag{6}$$

As  $s_i$  tends to  $s_c$  and  $t_i$  tends to  $t_c$ , the above equation becomes

$$h'(s_c) = \frac{T(t_c) \times T(s_c)}{\|\alpha(s_c) - \alpha(t_c)\| \kappa(s_c)} \frac{\kappa(s_c)}{\kappa(t_c)} = 0, \quad \text{since } T(t_c) \times T(s_c) = 0.$$

Meanwhile, we also obtain

$$\begin{aligned} h''(s_c) &= \left( T(t_c) \times T(s_c) \right) \frac{d}{ds} \left( \frac{1}{\|\alpha(h(s)) - \alpha(t)\| \kappa(h(s))} \frac{\kappa(s)}{\kappa(t)} \right) \Big|_{s=s_c} + \\ &\quad \left( \kappa(t_c) N(t_c) \frac{\kappa(s_c)}{\kappa(t_c)} \times T(s_c) + T(t_c) \times \kappa(s_c) N(s_c) h'(s_c) \right) \frac{1}{\|\alpha(s_c) - \alpha(t_c)\| \kappa(t_c)} \\ &= \frac{\kappa(s_c)}{\kappa(t_c) \|\alpha(s_c) - \alpha(t_c)\|} \quad \text{since } T(t_c) \times T(s_c) = 0 \text{ and } h'(s_c) = 0 \\ &\neq 0. \end{aligned}$$

Hence the sequence  $\{s_i\}$  converges quadratically.

**Theorem 5** *Let  $\mathcal{S}$  over  $(s_a, s_b)$  and  $\mathcal{T}$  over  $(t_a, t_b)$  be two segments of  $\alpha$  satisfying conditions (i)–(iii). Suppose  $\mathcal{S}$  and  $\mathcal{T}$  have exactly one common tangent with points of tangency  $s_c$  on  $\mathcal{S}$  and  $t_c$  on  $\mathcal{T}$ . Also suppose that a point traversing  $\mathcal{T}$  from  $t_a$  to  $t_b$  is moving towards  $s_c$  at the moment it reaches  $t_c$ . Then the two sequences  $s_0(=s_a), s_1, s_2, \dots$  and  $t_0(=t_a), t_1, t_2, \dots$  generated by `Common-Tangent` $(s_a, s_b, t_a, t_b)$  converge quadratically to  $s_c$  and  $t_c$ , respectively.*

Let us look at the time cost required for numerically solving (4) and (5) for the two points of tangency  $s_c$  and  $t_c$ . Suppose  $\omega$  binary bits of precision are required. Since  $h'(s_c) = 0$ , Taylor's expansion yields

$$\begin{aligned} s_{i+1} - s_c &= h(s_i) - h(s_c) \\ &= h''(s_c)(s_i - s_c)^2 + O((s_i - s_c)^3). \end{aligned}$$

So there exists some  $C > 0$  such that

$$\begin{aligned} |s_n - s_c| &\leq C|s_{n-1} - s_c|^2 \\ &\leq C \cdot C^2 |s_{n-2} - s_c|^{2^2} \\ &\vdots \\ &\leq C^{2^n - 1} |s_0 - s_c|^{2^n}. \end{aligned}$$

Without loss of generality, we assume that  $|s_0 - s_c| < 1$ . Otherwise, there exists some  $k$  such that  $|s_k - s_c| < 1$  since the sequence  $\{s_i\}$  is monotone. To reach a precision of  $2^{-\omega}$ , the number of iteration steps<sup>10</sup> must satisfy

$$n \geq \log \omega - \log \left( -\frac{\log C}{2} - \log |s_0 - s_c| \right).$$

In other words, the length of the sequence  $\{s_i\}$  (and thus of  $\{t_i\}$ ) needs to be  $O(\log \omega)$ . Obtaining  $s_{i+1}$  and  $t_{i+1}$  from  $s_i$  and  $t_i$  requires a call to each of the primitives `point-of-tangency` and `point-with-normal`, which take time  $O(\omega)$  as analyzed in Appendix A. Thus the number of low-level iteration steps in constructing the common tangent is  $O(\omega \log \omega)$ .

## 2.4 The Construction Algorithm

We now remove the two assumptions in Section 2.3 and describe how to compute common tangents of  $\mathcal{S}$  and  $\mathcal{T}$  when they satisfy conditions (i)–(iii). First, we use the procedure described in Section 2.2 to determine if a common tangent exists, and if so, classify the configuration of  $\mathcal{S}$  and  $\mathcal{T}$  as one of Figure 5(a), (b), (c), (d).

If  $\mathcal{S}$  and  $\mathcal{T}$  have two common tangents, then the procedure `Common-Tangent` will find the one whose tangency points are the closest to  $s_a$  and  $t_a$ . This removes the first assumption in Section 2.3.

To remove the second assumption in Section 2.3, we need to determine the order of the four endpoints passed on as arguments for calling `Common-Tangent`. This order determines which of the

---

<sup>10</sup>Here we make the assumption that  $-\frac{\log C}{2} - \log |s_0 - s_c| > 0$  as otherwise some  $s_k$  instead of  $s_0$  will always satisfy the inequality.

two pairs of opposite endpoints to start the iteration at, and, within the pair, which endpoint to update first. This would not be an issue for configuration (b) in Figure 5, where the two common tangents can be found using the procedure calls

$$\text{Common-Tangent}(s_a, s_b, t_a, t_b) \quad \text{and} \quad \text{Common-Tangent}(s_b, s_a, t_b, t_a)$$

regardless of the endpoint labeling.

When the configuration is one of (a), (c), (d), we look at the parallel tangent lines  $L_a$  and  $L'_a$  of  $\alpha$  at  $s_a$  and  $t_a$ , respectively. For instance, if the translation from  $L_a$  to  $L'_a$  is along the bending direction of  $\mathcal{S}$  then we use the procedure call  $\text{Common-Tangent}(t_a, t_b, s_a, s_b)$ . The iteration starts by updating from  $t_0 = t_a$  to  $t_1$ , the point of tangency of a tangent line to  $\mathcal{T}$  through  $s_a$ , then from  $s_0 = s_a$  to  $s_1$ , where  $N(s_1) + N(t_1) = 0$ , and so on.

## 2.5 Common Tangents of Two Curves

A common tangent of two curves (not necessarily closed) is incident on two points, one on each curve, whose normals are either the same or opposite to each other.

To construct all common tangents, we generate pairs of segments from different curves that satisfy conditions (i)–(iii). This is done as follows. Dissect both curves at their points of inflection and further split each segment with total curvature beyond  $[-\pi, \pi]$ . Enumerate all pairs of the resulting segments that are from different curves. On each pair, chop off sections if necessary until the endpoints have opposite normals as described by condition (ii). The above operations constitute the first three steps of preprocessing detailed in Section 5.

For each pair of segments construct their common tangents (if exists) as described in Section 2.4. Every common tangent found thus far will be incident on two points with opposite normals.

Next, we reverse the directions of all normals on one curve. Regenerate segment pairs that satisfy conditions (i)–(iii) and find the common tangents of every pair if they exist. Every common tangent found now is incident on two points whose normals were the same before the reversal.

## 3 Geometry at Antipodal Points

Recall from Section 1.2 that the curve  $\alpha$  is simple, closed, and twice continuously differentiable. Two points  $a$  and  $b$  on  $\alpha$  are *antipodal* if their inward normals are collinear and pointing at each other, or more precisely, if the following three conditions hold:

$$\begin{aligned} N(a) + N(b) &= 0, \\ N(a) \times (\alpha(b) - \alpha(a)) &= 0, \\ N(a) \cdot (\alpha(b) - \alpha(a)) &> 0. \end{aligned} \tag{7}$$

At least one pair of antipodal points exists on  $\alpha$ .<sup>11</sup> This is proved by Hong *et al.* [12] using a distance function

$$\chi(s, t) = (\alpha(s) - \alpha(t)) \cdot (\alpha(s) - \alpha(t)).$$

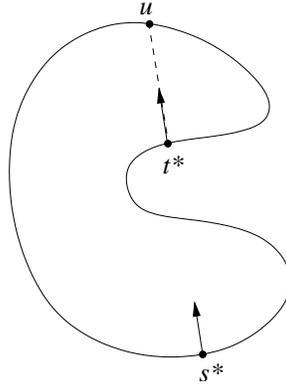
---

<sup>11</sup>In fact,  $\alpha$  need only be continuously differentiable for such existence.

Let  $s^*$  and  $t^*$  be the two points on  $\alpha$  that maximize  $\chi$ . Immediately,

$$\begin{aligned}\frac{\partial\chi(s^*, t^*)}{\partial s} &= 2T(s^*) \cdot (\alpha(s^*) - \alpha(t^*)) = 0; \\ \frac{\partial\chi(s^*, t^*)}{\partial t} &= -2T(t^*) \cdot (\alpha(s^*) - \alpha(t^*)) = 0.\end{aligned}$$

The unit normals  $N(s^*)$  and  $N(t^*)$  are thus collinear. In fact,  $N(s^*) + N(t^*) = 0$ . To see this, assume that  $N(s^*)$  and  $N(t^*)$  are in the same direction, and without loss of generality, in the direction of  $\alpha(t^*) - \alpha(s^*)$ . Let  $u$  be the intersection of  $\alpha$  with the ray extending  $N(t^*)$ , as shown in Figure 11. Then we see that  $\chi(s^*, u) > \chi(s^*, t^*)$ , contradicting that  $\chi(s^*, t^*)$  is the maximum. By a similar reasoning we obtain that  $N(s^*) \cdot (\alpha(t^*) - \alpha(s^*)) > 0$ . So  $s^*$  and  $t^*$  are antipodal. The



**Figure 11:** Diameter points must be antipodal.

maximum distance  $\sqrt{\chi(s^*, t^*)}$  is called the *diameter* of  $\alpha$ .

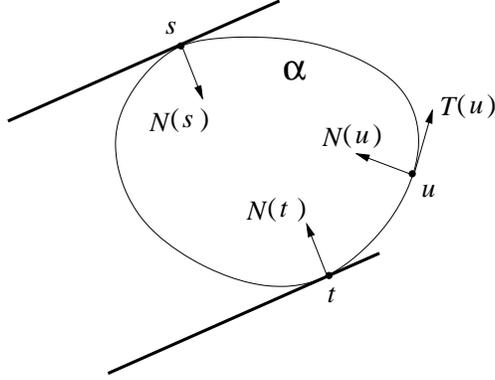
In [12], it was also established that at least two pairs of antipodal points exist when the curve is convex. Here we derive this result in a different way when  $\kappa = 0$  at no more than a finite number of points. Note that for every point  $s$ , there exists exactly one point  $t$  with the opposite normal, that is,  $N(t) = -N(s)$ . The function  $l(s) = \chi(s, t(s))$  attains at least one local maximum and one local minimum on the curve. They happen at critical points where  $l'(s)$  becomes zero, that is,

$$\left(T(s) - T(t)\frac{dt}{ds}\right) \cdot (\alpha(s) - \alpha(t)) = T(s) \left(1 + \frac{\kappa(s)}{\kappa(t)}\right) \cdot (\alpha(s) - \alpha(t)) = 0.$$

Given the convexity of  $\alpha$ , the above implies that  $T(s) \cdot (\alpha(t) - \alpha(s)) = 0$  and  $N(s) \cdot (\alpha(t) - \alpha(s)) > 0$ . Therefore  $s$  and  $t$  are antipodal. Also since there is at least one  $s$  value at which  $l(s)$  attains local minimum and at least one different  $s$  value at which  $l(s)$  attains local maximum,<sup>12</sup> there exist at least two different pairs of antipodal points. Imagine a pair of parallel lines that are tangent to the curve as in Figure 12. As the lines rotate along the curve boundary while maintaining the tangencies, their distance changes. Antipodal points correspond to the points of tangency where the two lines are (locally) the closest to or the furthest away from each other.

A closed convex simple curve can have as many pairs of antipodal points as possible. One trivial example is a circle whose curvature is constant everywhere. Every two points on the circle that

<sup>12</sup>Unless  $l(s)$  has the same value everywhere in which case every pair of  $s$  and  $t$  are antipodal.



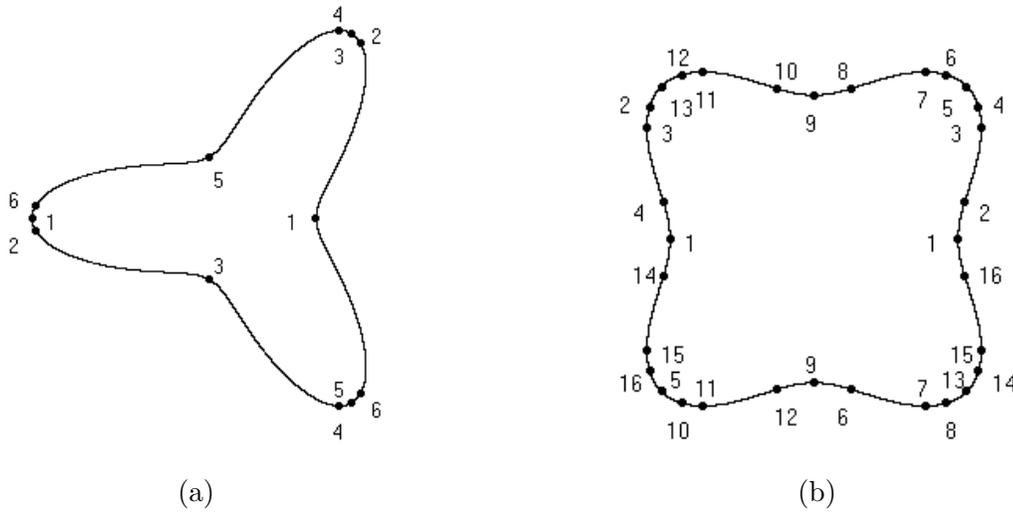
**Figure 12:** At least two pairs of antipodal points exist on a closed convex curve. Two parallel tangent lines rotate around the curve and are separated by locally maximum or minimum distance at antipodal points.

determine a diameter are antipodal. To exclude this degenerate (and trivial) case, we assume that  $\alpha$  has a non-constant curvature function.

An arbitrary number of pairs of antipodal points can still appear on curves with non-constant curvature. An example is a family of curves with convexities [39, pp. 183-185] defined in polar coordinates as

$$\rho(\phi) = \frac{p}{1 + \epsilon \cos m\phi}, \quad p > 0, \quad 0 < \epsilon < 1, \quad \text{integer } m > 1. \quad (8)$$

Figure 13 shows two curves in the family and the antipodal points on them. Such a curve has  $m$



**Figure 13:** Two curves from the family  $\rho(\phi) = \frac{p}{1 + \epsilon \cos m\phi}$  and their antipodal points: (a)  $p = 3, \epsilon = \frac{1}{2}, m = 3$  with 6 pairs antipodal points; (b)  $p = \frac{9}{2}, \epsilon = \frac{1}{5}, m = 4$  with 16 pairs of antipodal points. As many as  $\binom{2m}{2} - 2m$  pairs may exist.

axes of symmetry passing through  $m$  different pairs of antipodal points. These  $2m$  points, with extremum curvatures, are called the *vertices* of the curve. When  $m$  is even, both vertices on the same axis of symmetry have the maximum curvature or both have the minimum curvature. When

$m$  is odd, one of them has the maximum curvature and the other has the minimum curvature. When  $m$  is large enough and  $\epsilon$  is close enough to 1, the vertices can assume very high absolute curvature such that the neighborhoods of any two non-adjacent vertices will contain one distinct pair of antipodal points. So there can be as many as  $\binom{2m}{2} - 2m$  pairs of antipodal points.

The remainder of this section examines the differential geometry at antipodal points and classifies them into five types based on a novel definition of “antipodal angle”. The classification will be used in Section 4 for the development of an algorithm that finds all antipodal points.

### 3.1 Antipodal Angle

From now until Section 5 we focus our study on a pair of segments  $\mathcal{S}$  and  $\mathcal{T}$  of  $\alpha$  over  $(s_a, s_b)$  and  $(t_a, t_b)$  that meet conditions (i)–(iii) in Section 2 in addition to the two conditions below:

(iv) Neither  $s_a$  and  $t_a$  nor  $s_b$  and  $t_b$  are antipodal.

(v)  $N(s) \cdot (\alpha(t) - \alpha(s)) > 0$  for all  $s \in (s_a, s_b)$ .

Condition (iii) already ensures that a pair of antipodal points cannot appear on the same segment, which does not include the points  $s_a, s_b, t_a,$  or  $t_b$ . Condition (iv) states that the boundary points (not part of the two segments) are not antipodal. Condition (v) addresses that two opposite points may be antipodal only if their normals do not point away from each other.

Pairs of segments on  $\alpha$  that satisfy conditions (i)–(v) will be generated in the preprocessing phase described in Section 5.

Define the *antipodal angle*  $\theta(s)$  as the rotation angle from the normal  $N(s)$  to the vector  $\mathbf{r}(s) = \alpha(t) - \alpha(s)$ .<sup>13</sup> See Figure 14. Under condition (v),  $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ . By definition  $s$  and  $t$  are antipodal

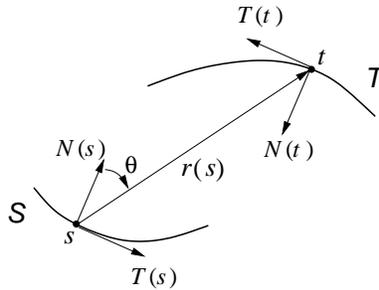


Figure 14: Antipodal angle  $\theta$ .

if and only if  $\theta(s) = 0$ .

Below we determine the rate of change of the antipodal angle  $\theta$ . First from

$$\|\mathbf{r}(s)\| = \sqrt{\mathbf{r}(s) \cdot \mathbf{r}(s)}$$

we calculate the derivative:

$$\frac{d}{ds} \|\mathbf{r}(s)\| = \frac{d}{ds} \sqrt{(\alpha(t) - \alpha(s)) \cdot (\alpha(t) - \alpha(s))}$$

<sup>13</sup>In [4], Blake and Taylor referred to it as the friction angle.

$$\begin{aligned}
&= \frac{\left(T(t)\frac{dt}{ds} - T(s)\right) \cdot \mathbf{r}(s)}{\|\mathbf{r}(s)\|} \\
&= -\left(\frac{dt}{ds} + 1\right) T(s) \cdot \frac{\mathbf{r}(s)}{\|\mathbf{r}(s)\|} && \text{by (3)} \\
&= -\left(\frac{\kappa(s)}{\kappa(t)} + 1\right) \cos\left(\frac{\pi}{2} + \theta\right) \\
&= \left(\frac{\kappa(s)}{\kappa(t)} + 1\right) \sin \theta.
\end{aligned}$$

From Figure 14 we also see that

$$\sin \theta = N(s) \times \frac{\mathbf{r}(s)}{\|\mathbf{r}(s)\|}. \quad (9)$$

Differentiate both sides of the above equation and substitute the expression for  $\frac{d}{ds}\|\mathbf{r}(s)\|$  in:

$$\begin{aligned}
\cos \theta \cdot \theta'(s) &= -\kappa(s)T(s) \times \frac{\mathbf{r}(s)}{\|\mathbf{r}(s)\|} \\
&\quad + N(s) \times \frac{\left(T(t)\frac{dt}{ds} - T(s)\right) \|\mathbf{r}(s)\| - \mathbf{r}(s)\frac{d}{ds}\|\mathbf{r}(s)\|}{\|\mathbf{r}(s)\|^2} \\
&= -\kappa(s) \cos \theta + N(s) \times \frac{-\frac{dt}{ds} - 1}{\|\mathbf{r}(s)\|} T(s) - \left(\frac{dt}{ds} + 1\right) \frac{\sin^2 \theta}{\|\mathbf{r}(s)\|} \\
&= -\kappa(s) \cos \theta + \frac{\frac{\kappa(s)}{\kappa(t)} + 1}{\|\mathbf{r}(s)\|} \cos^2 \theta.
\end{aligned}$$

On  $(s_a, s_b)$ ,  $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$  and  $\cos \theta > 0$ . Divide both sides of the above equation by  $\cos \theta$ :

$$\theta'(s) = -\kappa(s) + \frac{\cos \theta}{\|\mathbf{r}(s)\|} \left(\frac{\kappa(s)}{\kappa(t)} + 1\right). \quad (10)$$

### 3.2 Higher Order Antipodal Points

Suppose  $\alpha$  is at least  $k + 1 \geq 2$  times continuously differentiable. Two opposite points  $s^*$  and  $t^*$  on  $\alpha$  where  $\theta(s^*) = \theta'(s^*) = \dots = \theta^{(k-1)}(s^*) = 0$  but  $\theta^{(k)}(s^*) \neq 0$  are called *antipodal points of order k*. If  $k = 1$ , then  $s$  and  $t$  are also referred to as *simple antipodal points*.

When  $s^*$  and  $t^*$  are antipodal points of order 2, equation (10) reduces to

$$\|\mathbf{r}(s^*)\| = \frac{1}{\kappa(s^*)} + \frac{1}{\kappa(t^*)}. \quad (11)$$

Because  $\frac{1}{|\kappa(s^*)|}$  and  $\frac{1}{|\kappa(t^*)|}$  are the radii of the osculating circles<sup>14</sup> at  $s^*$  and  $t^*$ , respectively, these two circles are *tangent* to each other.

We are primarily interested in *finding simple antipodal points*, which will be simply referred to as antipodal points whenever no ambiguity arises.

---

<sup>14</sup>Definition of the osculating circle is given in Appendix C.

### 3.3 A Classification

At a pair of simple antipodal points  $s^*$  and  $t^*$ ,  $\theta(s^*) = 0$  and equation (10) becomes

$$\theta'(s^*) = -\kappa(s^*) + \frac{1}{\|\mathbf{r}(s^*)\|} \left( \frac{\kappa(s^*)}{\kappa(t^*)} + 1 \right). \quad (12)$$

Let us rearrange the terms in the above equation:

$$\left( 1 + \frac{\theta'(s^*)}{\kappa(s^*)} \right) \|\mathbf{r}(s^*)\| = \frac{1}{\kappa(s^*)} + \frac{1}{\kappa(t^*)}.$$

Let  $O_s$  be the center of curvature at  $s^*$  and  $O_t$  the center of curvature at  $t^*$ . Clearly,  $s^*, t^*, O_s, O_t$  are collinear. We now classify simple antipodal points into the following five types which are illustrated in Figure 15:

**Type A** :  $\alpha$  convex at both  $s^*$  and  $t^*$  and  $\frac{1}{\kappa(s^*)} + \frac{1}{\kappa(t^*)} < \|\mathbf{r}(s^*)\|$ . In this case  $\theta'(s^*) < 0$  and the order along  $\mathbf{r}(s^*)$  is  $s^*, O_s, O_t, t^*$ .

**Type B** :  $\alpha$  convex at both  $s^*$  and  $t^*$  and  $\frac{1}{\kappa(s^*)} + \frac{1}{\kappa(t^*)} > \|\mathbf{r}(s^*)\|$ . In this case  $\theta'(s^*) > 0$  and the order along  $\mathbf{r}(s^*)$  is  $s^*, O_t, O_s, t^*$ .

**Type C** :  $\alpha$  convex at one of  $s^*$  and  $t^*$  while concave at the other and  $\frac{1}{\kappa(s^*)} + \frac{1}{\kappa(t^*)} > \|\mathbf{r}(s^*)\|$ . The osculating circle at the concave antipodal point is entirely contained within the osculating circle at the convex antipodal point. In this case  $\theta'(s^*)$  and  $\kappa(s^*)$  have the same sign.

**Type D** :  $\alpha$  convex at one of  $s^*$  and  $t^*$  while concave at the other and  $\frac{1}{\kappa(s^*)} + \frac{1}{\kappa(t^*)} < \|\mathbf{r}(s^*)\|$ . The osculating circle at the concave antipodal point is not contained in the osculating circle at the convex antipodal point. In this case  $\theta'(s^*)$  and  $\kappa(s^*)$  have opposite signs.

**Type E** :  $\alpha$  concave at both  $s^*$  and  $t^*$ . That  $\frac{1}{\kappa(s^*)} + \frac{1}{\kappa(t^*)} < \|\mathbf{r}(s^*)\|$  always holds. In this case  $\theta'(s^*) > 0$ .

## 4 Computation of Antipodal Points

This section presents an algorithm that finds all simple antipodal points on two segments  $\mathcal{S}$  and  $\mathcal{T}$  of  $\alpha$  under conditions (i)–(v) given in Sections 2 and 3.1. Recall that  $\mathcal{S}$  and  $\mathcal{T}$  are defined over the intervals  $(s_a, s_b)$  and  $(t_a, t_b)$ , respectively. The algorithm works by alternatively marching on the two segments and bisecting them. Its behavior is determined by the signs of the antipodal angles at the endpoints  $s_a$  and  $s_b$  as well as by the convexity of the segments.

Table 1 gives the pseudocode of the algorithm as a procedure `Antipodal-Points` and its two subroutines `Bisect` and `March`. The procedure `Antipodal-Points` invokes either `Bisect` or `March` according to whether the signs of the antipodal angles  $\theta(s_a)$  and  $\theta(s_b)$  are the same or not. The subroutine `Bisect` performs numerical bisection to find a pair of antipodal points. This will be described in Section 4.1. The subroutine `March` searches for antipodal points in a march starting at one pair of opposing endpoints of  $\mathcal{S}$  and  $\mathcal{T}$ . It employs one of two different strategies, detailed in Sections 4.2.1 and 4.2.2, respectively, depending on whether one or both of the segments are convex.

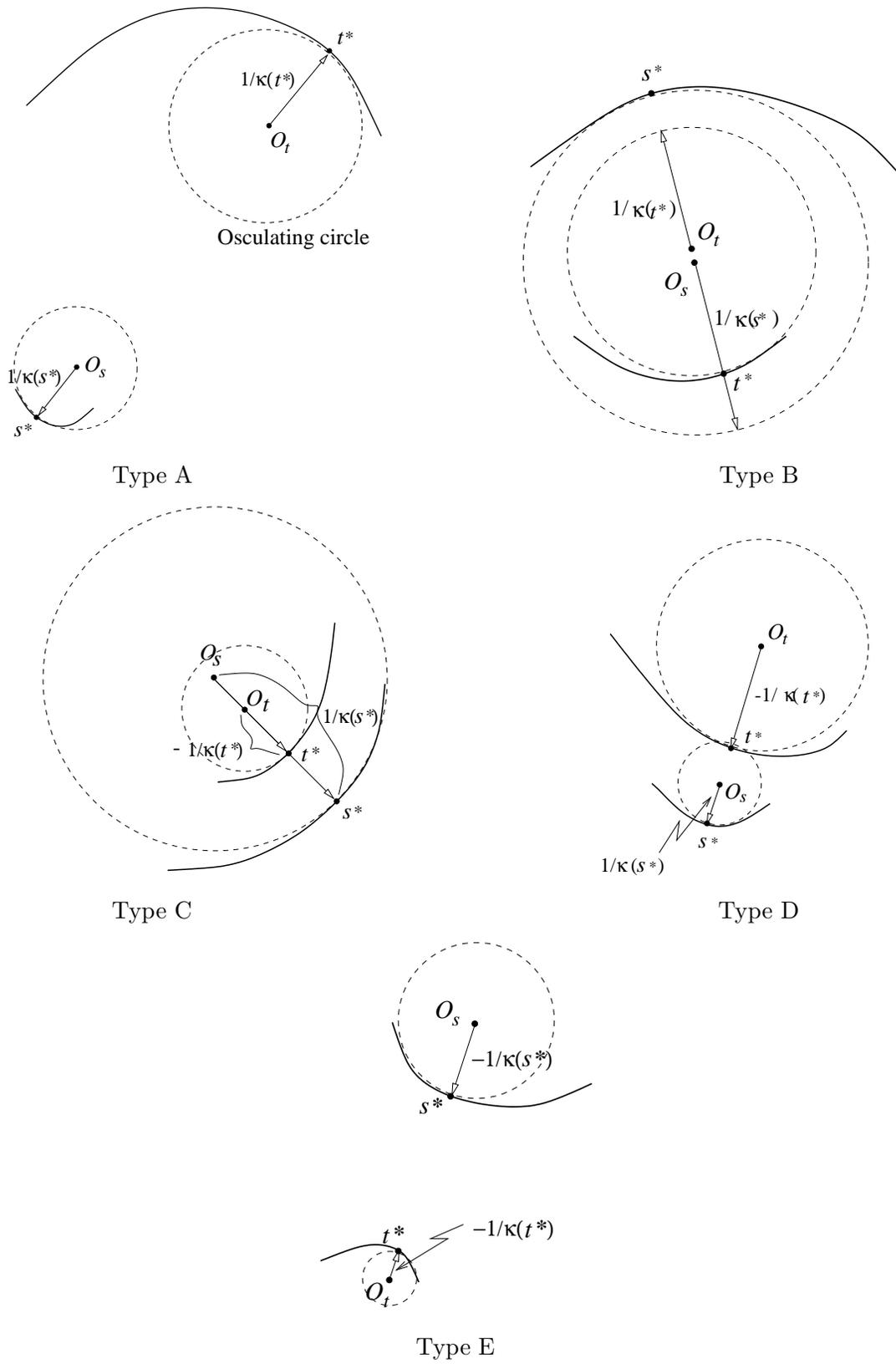


Figure 15: Five types of simple antipodal points.

```

Antipodal-Points( $s_a, s_b, t_a, t_b$ )
1  if  $\theta(s_a) \cdot \theta(s_b) < 0$ 
2    then Bisect( $s_a, s_b, t_a, t_b$ )
3    else March( $s_a, s_b, t_a, t_b$ )

Bisect( $s_a, s_b, t_a, t_b$ )
1  bisect the segments as described in Section 4.1
2  output the found pair of antipodal points ( $s^*, t^*$ )
3  if  $\mathcal{S}$  or  $\mathcal{T}$  is convex
4    then March( $s_a, s^* - \epsilon, t_a, t^* - \delta_1$ )
5        March( $s^* + \epsilon, s_b, t^* + \delta_2, t_b$ )

March( $s_a, s_b, t_a, t_b$ )
1  if both  $\mathcal{S}$  and  $\mathcal{T}$  are convex
2    then march on the segments as described in Section 4.2.1
3    else march on the segments as described in Section 4.2.2
4  if two antipodal points  $s^*$  and  $t^*$  are found
5    then output ( $s^*, t^*$ )
6        if the march started at  $s_b$  and  $t_b$ 
7            then Bisect( $s_a, s^* - \epsilon, t_a, t^* - \delta_1$ )
8            else Bisect( $s^* + \epsilon, s_b, t^* + \delta_2, t_b$ )

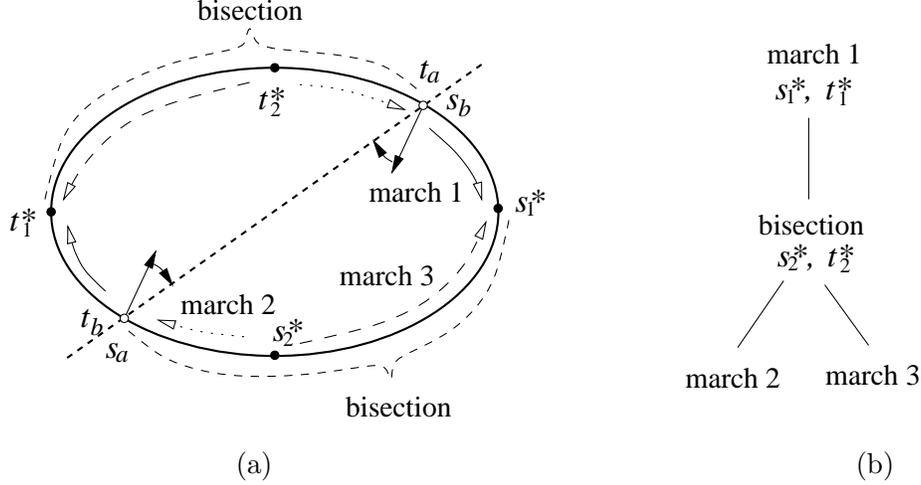
```

**Table 1:** The procedure **Antipodal-Points** finds all antipodal points (up to numerical precision) on a pair of segments over  $(s_a, s_b)$  and  $(t_a, t_b)$ , respectively. The subroutines **March** and **Bisect** call each other recursively. In the pseudocode,  $\epsilon > 0$  is a small constant,  $t^* - \delta_1$  is the opposite point of  $s^* - \epsilon$ , and  $t^* + \delta_2$  is the opposite point of  $s^* + \epsilon$ .

The subroutines **Bisect** and **March** call each other recursively to partition the two segments into smaller pieces and examine them. The recursive calls always terminate at the subroutine **March** unless  $\mathcal{S}$  and  $\mathcal{T}$  are both concave.

Whenever the subroutine **Bisect** is invoked, the two antipodal angles  $\theta(s_a)$  and  $\theta(s_b)$  have opposite signs. Clearly, this is true in line 2 of **Antipodal-Points**. This is also true in lines 7–8 of **March**, as will be seen in Sections 4.2.1 and 4.2.2. So the subroutine guarantees to find one pair of antipodal points. In the case where  $\mathcal{S}$  and  $\mathcal{T}$  are concave, at most one pair of antipodal points exists. This fact will be established in Section 4.1.

Figure 16 illustrates the working of **Antipodal-Points** on an ellipse. The ellipse is equally divided into two convex segments over  $(s_a, s_b)$  and  $(t_a, t_b)$ , respectively, where  $\alpha(s_a) = \alpha(t_b)$ ,  $\alpha(s_b) = \alpha(t_a)$ , and  $N(s_a) = -N(t_a)$ . Here  $\theta(s_a) < 0$  and  $\theta(s_b) < 0$ . The subroutine **March** performs a clockwise march starting at  $s_b$  and  $t_b$  in line 2 of its pseudocode and finds the first pair of antipodal points  $s_1^*$  and  $t_1^*$ . For small enough  $\epsilon > 0$ ,  $\theta(s_1^* - \epsilon) > 0$ . In line 7, the subroutine **Bisect** is invoked on  $(s_a, s_1^* - \epsilon)$  and  $(t_a, t_1^* - \delta_1)$ , where  $N(t_1^* - \delta_1) = -N(s_1^* - \epsilon)$ . This subroutine finds a second antipodal pair  $s_2^*$  and  $t_2^*$  in its line 1. Finally, in lines 4–5 of **Bisect**, a march is invoked on the pair  $(s_a, s_2^* - \epsilon)$  and  $(t_a, t_2^* - \delta_2)$  and another march on the pair  $(s_2^* + \epsilon, s_1^* - \epsilon)$  and  $(t_2^* + \delta_3, t_1^* - \delta_1)$ . No more antipodal points are found. Figure 16 (b) gives the recursion tree for



**Figure 16:** (a) Two pairs of antipodal points  $(s_1^*, t_1^*)$  and  $(s_2^*, t_2^*)$  on an ellipse; and (b) the recursion tree for finding them. Here  $\alpha(s_b) = \alpha(t_a)$  and  $N(s_a) = -N(t_a)$ .

the example.

#### 4.1 Opposite Angle Signs at Endpoints

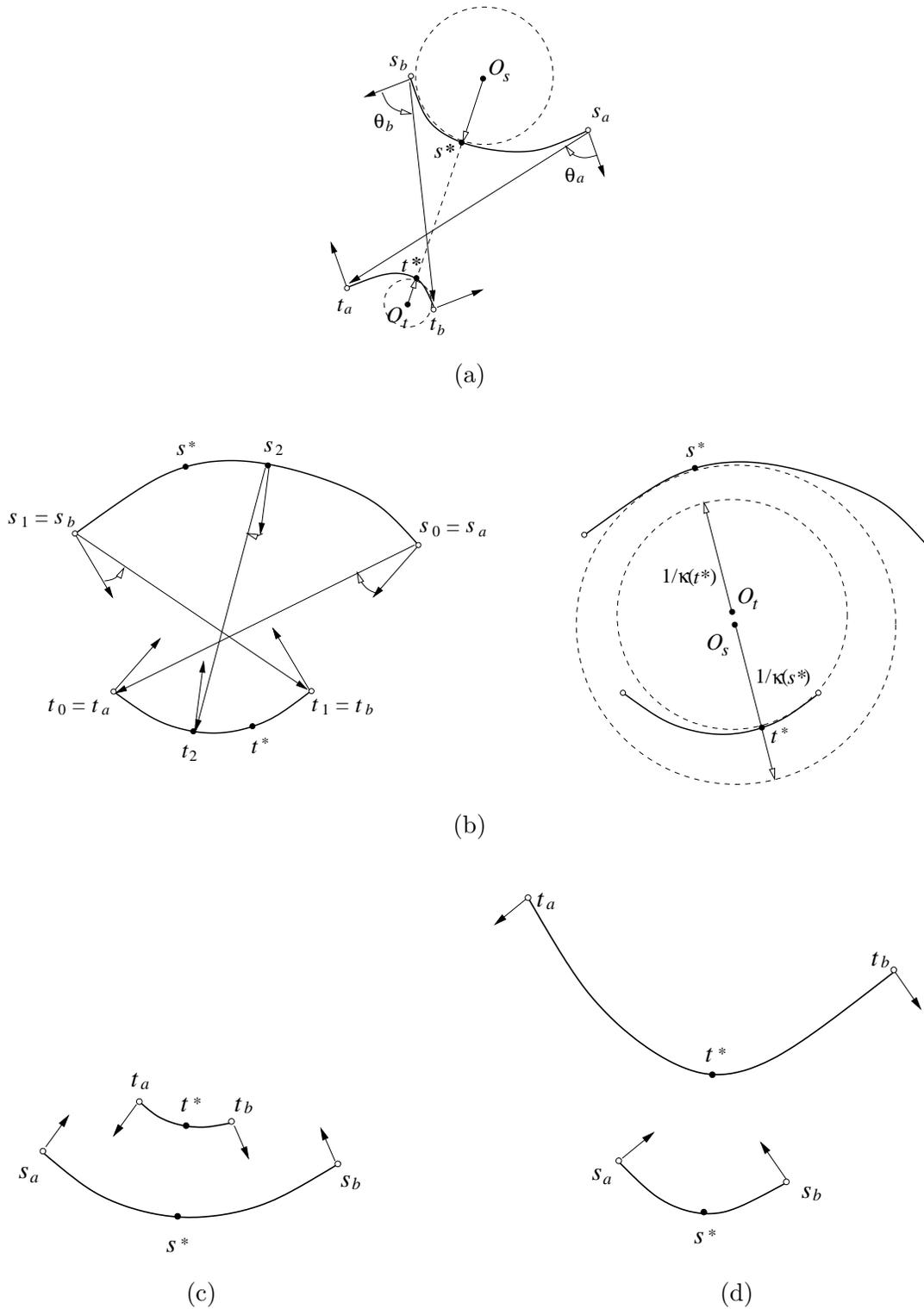
The subroutine `Bisect` is invoked when the antipodal angles  $\theta(s_a)$  and  $\theta(s_b)$  have different signs. At least one pair of antipodal points exists. The bisection method [33, pp. 261–263] guarantees to find one such pair, say  $(s^*, t^*)$ . Suppose  $\theta(s_a) < 0$  and  $\theta(s_b) > 0$ . Bisection starts by setting  $(s_0, t_0) \leftarrow (s_a, t_a)$  and  $(s_1, t_1) \leftarrow (s_b, t_b)$ . Then it evaluates  $s_2 \leftarrow \frac{s_0 + s_1}{2}$  and finds  $t_2 \in (t_0, t_1)$  such that  $N(t_2) = -N(s_2)$ . If  $\theta(s_2) > 0$ , set  $(s_1, t_1) \leftarrow (s_2, t_2)$ ; otherwise, set  $(s_0, t_0) \leftarrow (s_2, t_2)$ . Repeat the above steps until  $\theta(s_2)$  approaches 0. For the evaluation of  $\theta(s_2)$ ,<sup>15</sup> a primitive call `point-with-normal`( $t_0, t_1, -N(s_2)$ ) is made to find  $t_2$ .

Three cases arise based on the convexities of  $\mathcal{S}$  and  $\mathcal{T}$ .

- Both  $\mathcal{S}$  and  $\mathcal{T}$  are concave. So  $\kappa(s) < 0$  and  $\kappa(t) < 0$ . It follows from (10) that  $\theta'(s) > 0$ . The antipodal angle  $\theta$  increases monotonically from  $s_a$  to  $s_b$ . A unique pair of antipodal points (of Type E) exists if  $\theta(s_a) < 0$  and  $\theta(s_b) > 0$ . Otherwise, no antipodal points exist. Figure 17(a) shows an example.
- Both segments are convex. Either  $\theta'(s^*) < 0$  or  $\theta'(s^*) > 0$  holds. The antipodal points are of either Type A or Type B accordingly. An example of Type B antipodal points is shown in Figure 17(b).
- One of segment is convex and the other is concave. Either  $\theta'(s^*) > 0$  or  $\theta'(s^*) < 0$  holds. So  $s^*$  and  $t^*$  are of either Type C or Type D as shown in Figure 17(c) and (d), respectively.

Each step of bisection halves the interval, increasing the precision on the estimates of  $s^*$  and  $t^*$  by one binary bit. A precision requirement of  $\omega$  binary bits can thus be achieved in  $\omega + \log(s_b - s_a) =$

<sup>15</sup>In implementation, we do not need to compute  $\theta(s)$  explicitly. Instead we evaluate the cross product  $N(s) \times \mathbf{r}(s)$ . That  $-\pi/2 < \theta < \pi/2$  guarantees a one-to-one correspondence between  $\theta$  and  $\sin \theta = N(s) \times \frac{\mathbf{r}(s)}{\|\mathbf{r}(s)\|}$ . For instance, to test the condition  $\theta(s_2) > 0$ , we would actually test the condition  $N(s_2) \times \mathbf{r}(s_2) > 0$ .



**Figure 17:** Antipodal points  $s^*$  and  $t^*$  found using bisection, where  $O_s$  and  $O_t$  are the centers of the osculating circles at  $s^*$  and  $t^*$ , respectively. (a) two concave segments; (b) two convex segments with  $\theta'(s^*) > 0$ ; (c) a convex segment and a concave segment with  $\theta'(s^*) > 0$ ; (d) a convex segment and a concave segment with  $\theta'(s^*) < 0$ .

$\Theta(\omega)$  steps. Every bisection step involves a call to the primitive `point-with-normal` to find  $t_i$  from  $s_i$ . This requires  $O(\omega)$  numerical steps. So the total number of low-level numerical steps is  $O(\omega^2)$ .<sup>16</sup>

## 4.2 Same Angle Sign at Endpoints

When the antipodal angles  $\theta(s_a)$  and  $\theta(s_b)$  have the same sign, at least one of  $\mathcal{S}$  and  $\mathcal{T}$  must be convex in order to have antipodal points. Sections 4.2.1 and 4.2.2 will present two different “marching” strategies to find one pair of antipodal points if exists. Together they implement the subroutine `March` in Table 1.

### 4.2.1 Two Convex Segments

Since  $\kappa(s) > 0$  over  $\mathcal{S}$  and  $\kappa(t) > 0$  over  $\mathcal{T}$ , we cannot determine the sign of  $\theta'(s)$ . A march on  $\mathcal{S}$  and  $\mathcal{T}$  is performed in the search for antipodal points, based on the following fact.

**Proposition 6** *When  $\mathcal{S}$  and  $\mathcal{T}$  are convex, the vector  $\mathbf{r}(s)$  rotates counterclockwise as  $s$  increases from  $s_a$  to  $s_b$ .*

**Proof** We need only show that  $\frac{d\mathbf{r}}{ds} \times \mathbf{r} < 0$ . Differentiate the vector  $\mathbf{r}$ :

$$\begin{aligned} \frac{d\mathbf{r}}{ds} &= \frac{d}{ds}(\boldsymbol{\alpha}(t) - \boldsymbol{\alpha}(s)) \\ &= T(t) \left( 1 + \frac{\kappa(s)}{\kappa(t)} \right). \end{aligned}$$

Hence  $\frac{d\mathbf{r}}{ds}$  is in the direction of  $T(t)$  since  $\kappa(s), \kappa(t) > 0$ . Meanwhile, from condition (v) that  $\mathbf{r}(s) \cdot N(s) > 0$  it follows that

$$\begin{aligned} T(t) \times \mathbf{r}(s) &= -T(s) \times \mathbf{r}(s) \\ &= -N(s) \cdot \mathbf{r}(s) \\ &< 0. \end{aligned}$$

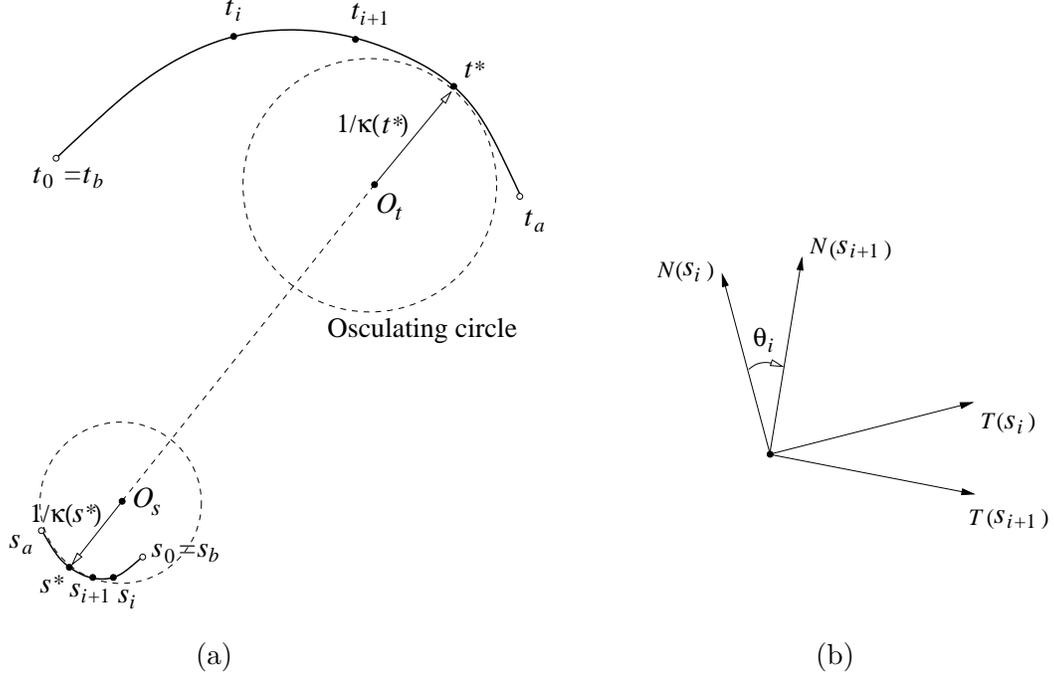
Therefore  $\frac{d\mathbf{r}(s)}{ds} \times \mathbf{r}(s) < 0$ . □

Figure 18 illustrates the march when  $\theta(s_a) < 0$  and  $\theta(s_b) < 0$ . It starts with  $s$  and  $t$  at  $s_0 = s_b$  and  $t_0 = t_b$ , respectively. By Proposition 6, the vector  $\mathbf{r}(s)$  rotates clockwise as  $s$  decreases to  $s_a$ . In the  $i$ th iteration, move  $s$  from  $s_i$  to  $s_{i+1}$  at which the normal is parallel to the vector  $\mathbf{r}(s_i)$ . If no such point  $s_{i+1}$  exists, stop. Otherwise, move  $t$  from  $t_i$  to  $t_{i+1}$  where  $N(t_{i+1}) + N(s_{i+1}) = 0$ . The iteration continues until  $s_i$  and  $t_i$  converge to a pair of antipodal points, as in the figure, or no  $s_{i+1}$  can be found (that is,  $s_a$  has been reached), in which case no antipodal points exist.

**Lemma 7** *In the case  $\theta(s_a) < 0$  and  $\theta(s_b) < 0$  of marching,  $s_i > s_{i+1}$  and every  $s \in [s_{i+1}, s_i]$  satisfies  $\theta(s) < 0$  for all  $i \geq 0$ .*

---

<sup>16</sup>In fact, we may also apply Newton-Raphson’s method to find  $s^*$  and  $t^*$ , where the derivative  $\theta'$  is evaluated according to (10). If Newton-Raphson fails, then we fall back on bisection. However, evaluation of the derivative of antipodal angle might be expensive enough to offset the quadratic convergence of Newton-Raphson.



**Figure 18:** Geometry at two antipodal points  $s^*$  and  $t^*$  with positive curvatures:  $\theta'(s^*) < 0$  and  $s^*$  is closer to the center of curvature  $O_s$  at  $s^*$  than to the center of curvature  $O_t$  at  $t^*$ . Here  $\theta_i = \theta(s_i)$ .

**Proof** We use induction. That  $\theta(s_0) = \theta(s_b) < 0$  follows directly from the initial condition. Suppose  $\theta(s_i) < 0$ . The normal  $N(s)$  rotates clockwise as  $s$  decreases from  $s_i$ . Also since  $N(s_i) \times \mathbf{r}(s_i) < 0$  and the normal  $N(s_{i+1})$ , if  $s_{i+1}$  exists, is in the direction of  $\mathbf{r}(s_i)$ , we know that

$$\begin{aligned} s_{i+1} &< s_i; \\ N(s) \times \mathbf{r}(s_i) &< 0, \quad \text{for all } s \in (s_{i+1}, s_i). \end{aligned} \quad (13)$$

Under condition (v),  $\mathbf{r}(s_{i+1}) \cdot N(s_{i+1}) > 0$ ; hence  $\mathbf{r}(s_{i+1}) \cdot \mathbf{r}(s_i) > 0$ . By Proposition 6,  $\mathbf{r}(s)$  rotates clockwise from  $s_i$  to  $s_{i+1}$ . It then follows that

$$\mathbf{r}(s_i) \times \mathbf{r}(s) < 0, \quad \text{for all } s \in [s_{i+1}, s_i]. \quad (14)$$

Combining inequalities (13) and (14) with condition (v), we have

$$N(s) \times \mathbf{r}(s) < 0, \quad \text{for all } s \in [s_{i+1}, s_i].$$

Thus  $\theta(s) < 0$  for all  $s \in [s_{i+1}, s_i]$ . □

Lemma 7 states that the sequence  $s_0, s_1, s_2, \dots$  generated above according to the equation

$$N(s_{i+1}) \times \mathbf{r}(s_i) = 0 \quad (15)$$

is *strictly decreasing* and no antipodal point exists on  $[s_i, s_b) = \cup_{k=1}^i [s_k, s_{k-1})$  for all  $i > 0$ . The sequence  $\{s_i\}$  is bounded from below by  $s^*$ , the closest antipodal point to  $s_b$  if exists. So it must converge to some  $\xi \in (s_a, s_b)$  where  $N(\xi) \times \mathbf{r}(\xi) = 0$ . Hence  $\xi$  is an antipodal point and  $\xi = s^*$ .

Next, we determine the local convergence rate of the sequence  $\{s_i\}$ . We want to look at  $f'(s^*)$  first. For simplicity, denote the antipodal angle  $\theta(s_i)$  as  $\theta_i$ . By the definition of antipodal angle and also from that  $\mathbf{r}(s_i)$  and  $N(s_{i+1})$  are in the same direction we have

$$\sin \theta_i = N(s_i) \times N(s_{i+1}).$$

Differentiating both sides of the above equation yields

$$\begin{aligned} \cos \theta_i \theta'(s_i) &= -\kappa(s_i)T(s_i) \times N(s_{i+1}) + N(s_i) \times \left(-\kappa(s_{i+1})f'(s_i)T(s_{i+1})\right) \\ &= -\kappa(s_i) \sin\left(\frac{\pi}{2} + \theta_i\right) - \kappa(s_{i+1}) \sin\left(\theta_i - \frac{\pi}{2}\right) f'(s_i) \quad \text{from Figure 18(b)} \\ &= -\kappa(s_i) \cos \theta_i + \kappa(s_{i+1}) \cos \theta_i f'(s_i). \end{aligned}$$

Hence

$$\begin{aligned} f'(s_i) &= \frac{\theta'(s_i) + \kappa(s_i)}{\kappa(s_{i+1})}, \\ f'(s^*) &= \frac{\theta'(s^*)}{\kappa(s^*)} + 1 \\ &= \frac{\kappa(s^*) + \kappa(t^*)}{\kappa(s^*)\kappa(t^*)\|\mathbf{r}(s^*)\|} > 0, \quad \text{by (12)} \end{aligned} \tag{16}$$

where  $t^*$  on  $\mathcal{T}$  is the opposite point of  $s^*$ . The march starts at  $s_b$  where  $\theta(s_b) < 0$  and never passes  $s^*$ . So  $\theta'(s^*) < 0$  must hold in the non-degenerate case.<sup>17</sup> It then follows from (16) that  $f'(s^*) < 1$ . The march converges at linear rate to  $s^*$ .

When  $\theta(s_a) > 0$  and  $\theta(s_b) > 0$ , the march starts at  $s_a$  and  $t_a$  and moves toward  $s_b$  and  $t_b$ . The result on convergence still holds. Combining the above analysis and Lemma 7, we have established the correctness of the marching procedure.

**Theorem 8** *The following statements hold after the execution of line 2 in the subroutine **March**:*

1. *If no antipodal points exist on  $\mathcal{S}$  and  $\mathcal{T}$ , the march terminates at  $s_a$  and  $t_a$  in the case  $\theta(s_a) < 0$  and  $\theta(s_b) < 0$  or at  $s_b$  and  $t_b$  in the case  $\theta(s_a) > 0$  and  $\theta(s_b) > 0$ .*
2. *Otherwise, the march converges at linear rate to a pair of antipodal points  $s^*$  and  $t^*$  closest to the two starting endpoints. Furthermore,  $\theta'(s^*) < 0$  must hold. The antipodal points are of Type A.*

Given the linear convergence rate, the two sequences  $\{s_i\}$  and  $\{t_i\}$  must have length  $O(\omega)$  to achieve the precision of  $2^{-\omega}$  on the estimated  $s^*$  and  $t^*$ . The step from  $s_i$  to  $s_{i+1}$  and then to  $t_{i+1}$  involves two calls to the primitive **point-with-normal**, which takes  $O(\omega)$  steps. So the march takes  $O(\omega^2)$  numerical steps to obtain the pair of antipodal points. If no antipodal points exist, the march will reach the other pair of endpoints. The lengths of the sequences  $\{s_i\}$  and  $\{t_i\}$  in this case is independent of  $\omega$  and can be treated as constant.

---

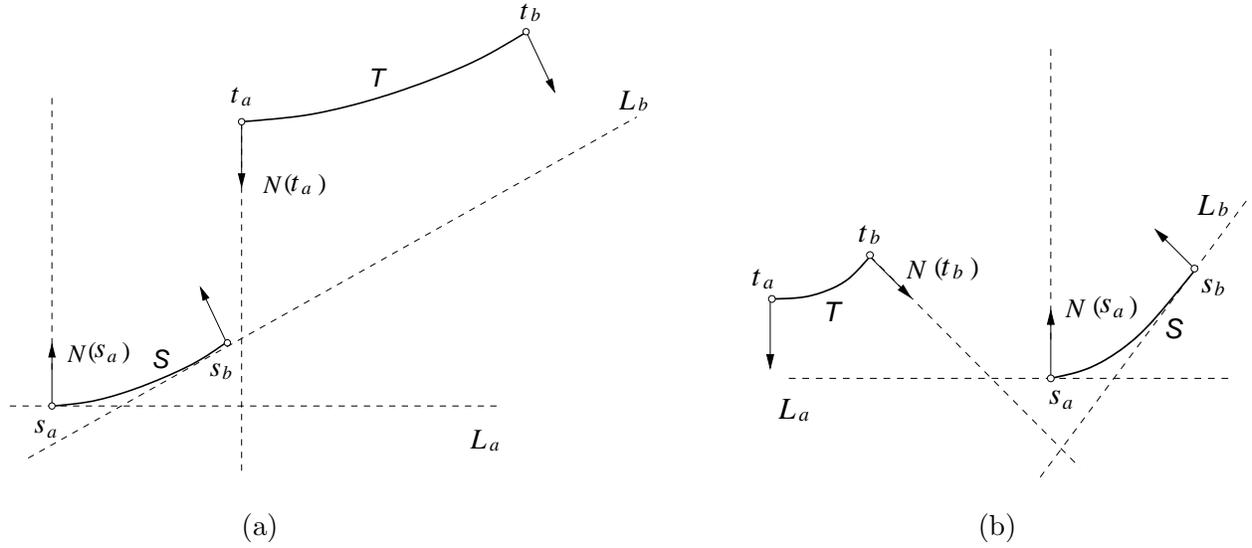
<sup>17</sup>That  $\theta'(s^*) = 0$  is the degenerate case, in which case the two antipodal points are of at least the second order.

### 4.2.2 A Convex Segment and A Concave Segment

We need only consider the case that  $\mathcal{S}$  is convex and  $\mathcal{T}$  is concave as the other case is symmetrical. First, we determine if one of the rays extending the normals  $N(t_a)$  and  $N(t_b)$  intersects  $\mathcal{S}$ . Under conditions (i)–(v) in Sections 2 and 3.1, testing if the ray extending  $N(t_a)$  at the point  $t_a$ , or simply called the *ray of  $N(t_a)$* , intersects  $\mathcal{S}$  can be done by checking whether the cross products  $(\alpha(s_a) - \alpha(t_a)) \times N(t_a)$  and  $(\alpha(s_b) - \alpha(t_a)) \times N(t_a)$  have different signs.

**Lemma 9** *Suppose  $\mathcal{S}$  is convex and  $\mathcal{T}$  is concave. Also suppose that the two antipodal angles  $\theta(s_a)$  and  $\theta(s_b)$  have the same sign. No antipodal points exist on  $\mathcal{S}$  and  $\mathcal{T}$  if neither the ray of  $N(t_a)$  nor the ray of  $N(t_b)$  intersects  $\mathcal{S}$ .*

**Proof** Without loss of generality, we assume that  $N(s_a)$  points vertically upward, as shown in Figure 19. Under condition (v),  $\mathcal{S}$  and  $\mathcal{T}$  must lie on the same sides of the two tangent lines  $L_a$



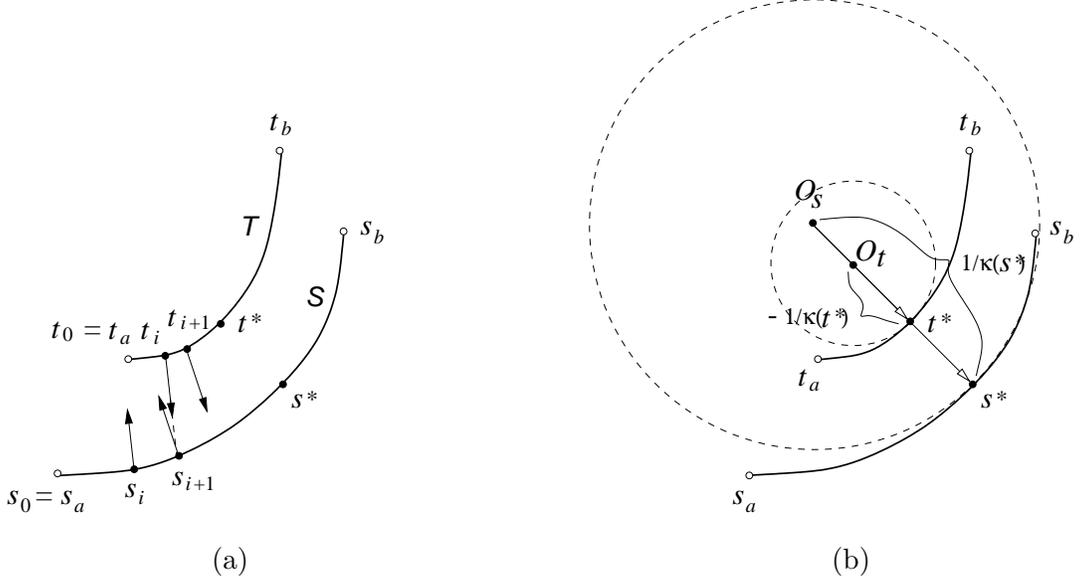
**Figure 19:** No antipodal points when neither the ray of  $N(t_a)$  nor the ray of  $N(t_b)$  intersects the segment  $\mathcal{S}$ .

and  $L_b$  of  $\mathcal{S}$  at  $s_a$  and  $s_b$ , respectively.

Suppose neither of the rays of  $N(t_a)$  and  $N(t_b)$  intersects  $\mathcal{S}$ . Because  $N(t_a)$  does not intersect  $\mathcal{S}$ ,  $t_a$  is either to the right of  $\mathcal{S}$  or to its left. If  $t_a$  is to the right, condition (v) requires that the segment  $\mathcal{T}$  do not cross the line containing  $N(t_a)$  into its left. So  $\mathcal{T}$  lies entirely to the right of the segment  $\mathcal{S}$ , as shown in Figure 19(a). But all normals on  $\mathcal{S}$  point to the left. Thus no antipodal points exist.

If  $t_a$  is to the left of  $\mathcal{S}$ , then  $\theta(s_a) > 0$ . Because  $\theta(s_b)$  has the same sign,  $\theta(s_b) > 0$ . Then  $\mathcal{S}$  and  $\mathcal{T}$  must lie on different sides of the line containing  $N(t_b)$  as shown in Figure 19(b). Apparently, they cannot have antipodal points either.  $\square$

When either the ray of  $N(t_a)$  or the ray of  $N(t_b)$  intersects  $\mathcal{S}$ , we carry out the following march. Start at  $s_0 = s_a$  and  $t_0 = t_a$  if the ray of  $N(t_a)$  intersects  $\mathcal{S}$  as illustrated in Figure 20, or at  $s_0 = s_b$  and  $t_0 = t_b$  if the ray of  $N(t_b)$  intersects  $\mathcal{S}$ . In each round,  $s_{i+1}$  is generated as the intersection of the ray of  $N(t_i)$  and  $\mathcal{S}$  and  $t_{i+1}$  is generated as its opposite point.



**Figure 20:** (a) Marching toward two antipodal points  $s^*$  and  $t^*$  when the ray of  $N(t_a)$  intersects  $\mathcal{S}$ : the normal at  $t_{i+1}$  is opposite to that at  $s_{i+1}$ , where the ray of  $N(t_i)$  intersects  $\mathcal{S}$ . (b) The osculating circle at  $t^*$  is contained entirely within the osculating circle at  $s^*$ .

**Lemma 10** *Suppose  $\mathcal{S}$  is convex and  $\mathcal{T}$  is concave and the two antipodal angles at the endpoints  $s_a$  and  $s_b$  of  $\mathcal{S}$  have the same sign. Also suppose the ray of  $N(t_a)$  intersects  $\mathcal{S}$ . In the march,  $s_i < s_{i+1}$  and no antipodal points exist in  $(s_i, s_{i+1}]$  and  $(t_i, t_{i+1}]$  for all  $i \geq 0$ .*

**Proof** The conditions imply that  $\theta(s_a) < 0$ . We use induction to prove that, for all  $i \geq 0$ ,  $\theta(s_i) < 0$  and every point  $s \in (s_i, s_{i+1}]$  and its opposite point  $t \in (t_i, t_{i+1}]$  cannot be antipodal. Obviously,  $s_0 = s_a$  and  $t_0 = t_a$  are not antipodal. Suppose  $s_i$  and  $t_i$  are not antipodal. Since  $\mathcal{S}$  is convex and  $0 < \Phi(s_a, s_b) \leq \pi$ , we have that  $N(u) \times N(v) > 0$  for any  $u, v \in (s_a, s_b)$  with  $u < v$ . If  $s_{i+1}$  exists on  $\mathcal{S}$ , then  $s_{i+1} > s_i$  holds due to that  $\theta(s_i) < 0$  and that  $s_{i+1}$  is the intersection of the ray of  $N(t_i)$  with  $\mathcal{S}$ . Therefore for all  $s \in (s_i, s_{i+1}]$

$$N(s) \times N(s_i) < 0 \quad \text{and} \quad N(s) \times (\alpha(t_i) - \alpha(s_{i+1})) < 0, \quad (17)$$

since  $\alpha(t_i) - \alpha(s_{i+1})$  is in the direction of the normal  $N(s_i)$ . Meanwhile, the point  $\alpha(s)$  lies to the left of the line through  $\alpha(t_i)$  and  $\alpha(s_{i+1})$  while  $\alpha(t)$  lies to its right; hence

$$(\alpha(t_i) - \alpha(s_{i+1})) \times (\alpha(t) - \alpha(s)) < 0. \quad (18)$$

Conditions (17) and (18) plus condition (v) imply that  $N(s) \times (\alpha(t) - \alpha(s)) < 0$ . Hence  $s$  and  $t$  cannot be antipodal for all  $s \in (s_i, s_{i+1}]$ . In particular,  $s_{i+1}$  and  $t_{i+1}$  are not antipodal and  $\theta(s_{i+1}) < 0$ .  $\square$

Following Lemma 10, the sequence  $\{s_i\}$  defined by

$$(\alpha(t_i) - \alpha(s_{i+1})) \times N(s_i) = 0 \quad (19)$$

is strictly increasing. And no antipodal points exist on  $(s_a, s_i]$  for all  $i > 0$ . If there exists at least one antipodal point on  $\mathcal{S}$ , the sequence  $\{s_i\}$  will converge to the closest such point  $s^*$  from  $s_a$ .

Next, we study the local convergence rate of the sequence. Equation (19) defines an iteration function  $g$  where  $s_{i+1} = g(s_i)$ . Differentiate this equation with respect to  $s_i$ :

$$\left(T(t_i) \frac{dt}{ds}(s_i) - T(s_{i+1})g'(s_i)\right) \times N(s_i) + (\alpha(t_i) - \alpha(s_{i+1})) \times (-\kappa(s_i)T(s_i)) = 0.$$

The above equation reduces to

$$-\frac{dt}{ds}(s_i) - g'(s_i)T(s_{i+1}) \times N(s_i) + \kappa(s_i) \|\alpha(t_i) - \alpha(s_{i+1})\| = 0,$$

from which we obtain

$$g'(s_i) = \frac{\kappa(s_i) \|\alpha(t_i) - \alpha(g(s_i))\| - \frac{\kappa(s_i)}{\kappa(t_i)}}{T(g(s_i)) \times N(s_i)},$$

$$g'(s^*) = \kappa(s^*) \|\alpha(t^*) - \alpha(s^*)\| - \frac{\kappa(s^*)}{\kappa(t^*)}.$$

Because  $\kappa(s^*) > 0$  and  $\kappa(t^*) < 0$ ,  $g'(s^*) > 0$ . Because  $\theta(s_i) < 0$  and  $s_{i+1} > s_i$ , for  $i = 0, 1, \dots$ ,  $\theta'(s^*) > 0$ ; in other words,

$$-\kappa(s^*) + \frac{1}{\|\mathbf{r}(s^*)\|} \left( \frac{\kappa(s^*)}{\kappa(t^*)} + 1 \right) > 0. \quad (20)$$

in the non-degenerate case. This implies that  $g'(s^*) < 1$ . So we have that  $0 < g'(s^*) < 1$ . Geometrically, the osculating circle at  $s^*$  contains the osculating circle at  $t^*$  in its interior, as shown in Figure 20(b).

Similar analysis can be performed for the case where the ray of  $N(t_b)$  intersects  $\mathcal{S}$ .

**Theorem 11** *Suppose  $\mathcal{S}$  convex and  $\mathcal{T}$  concave. No antipodal points exist on  $\mathcal{S}$  and  $\mathcal{T}$  if neither the ray of  $N(t_a)$  nor the ray of  $N(t_b)$  intersects  $\mathcal{S}$ . Otherwise, the following statements hold after the execution of line 3 in the subroutine **March**:*

1. *If no antipodal points exist on  $\mathcal{S}$  and  $\mathcal{T}$ , then the march described above will terminate at the other endpoints of  $\mathcal{S}$  and  $\mathcal{T}$ .*
2. *Otherwise, the march will converge at linear rate to a pair of antipodal points  $s^*$  and  $t^*$  closest to the two starting endpoints. Furthermore,  $\theta'(s^*) > 0$ . The antipodal points are of Type C.*

The linear convergence rate of the sequence  $\{s_i\}$  defined by (19) implies that the length of the sequence must be  $O(\omega)$  to attain  $\omega$  binary bits of precision on the estimated  $s^*$ . Each iteration step from  $s_i$  and  $t_i$  to  $s_{i+1}$  and  $t_{i+1}$  involves a call each to the primitives **point-on-ray** and **point-with-normal**. Such a step requires  $O(\omega)$  numerical steps. Therefore the number of low-level iteration steps in obtaining  $s^*$  and  $t^*$  is  $O(\omega^2)$ .

### 4.3 Correctness and Running Time

Combining the results from Sections 4.1 and 4.2, the subroutines `Bisect` and `March`, we now establish the correctness of the procedure `Antipodal-Points`.

**Theorem 12** *The procedure `Antipodal-Points` finds all pairs of antipodal points on any two curve segments satisfying conditions (i)–(v) if the constant  $\epsilon$  is chosen small enough in the subroutines `Bisect` and `March`.*

**Proof** Consider a pair of such segments  $\mathcal{S}$  and  $\mathcal{T}$  defined over  $(s_a, s_b)$  and  $(t_a, t_b)$ , respectively. It suffices to show that the subroutines `Bisect` and `March` find all pairs of antipodal points under the conditions  $\theta(s_a) \cdot \theta(s_b) < 0$  and  $\theta(s_a) \cdot \theta(s_b) > 0$ , respectively. We use induction on the number  $k$  of pairs of antipodal points.

The base case  $k = 1$  follows from the correctness of bisection and marching in finding one pair of antipodal points when exists. This has been established in Sections 4.1, 4.2.1, and 4.2.2.

A call to the subroutine `Bisect` always finds a pair of antipodal points, say,  $s^*$  and  $t^*$ , which splits  $\mathcal{S}$  and  $\mathcal{T}$  into two pairs of segments. Assume that  $\epsilon$  is small enough such that the interval  $(s^* - \epsilon, s^* + \epsilon)$  contains no antipodal point other than  $s^*$ . The subroutine shortens the two segments on  $\mathcal{S}$  to be over  $(s_a, s^* - \epsilon)$  and  $(s^* + \epsilon, s_b)$ , respectively. The two segments on  $\mathcal{T}$  are shortened correspondingly such that the resulting two new pairs of segments satisfy conditions (i)–(v). Numerical bisection ensures that the antipodal angles  $\theta(s_a)$  and  $\theta(s^* - \epsilon)$  have the same sign, and so do the antipodal angles  $\theta(s^* + \epsilon)$  and  $\theta(s_b)$ . Each new pair contains at most  $k - 1$  pairs of antipodal points. By induction, the two recursive calls to the subroutine `March` on lines 4–5 will find the remaining antipodal points.

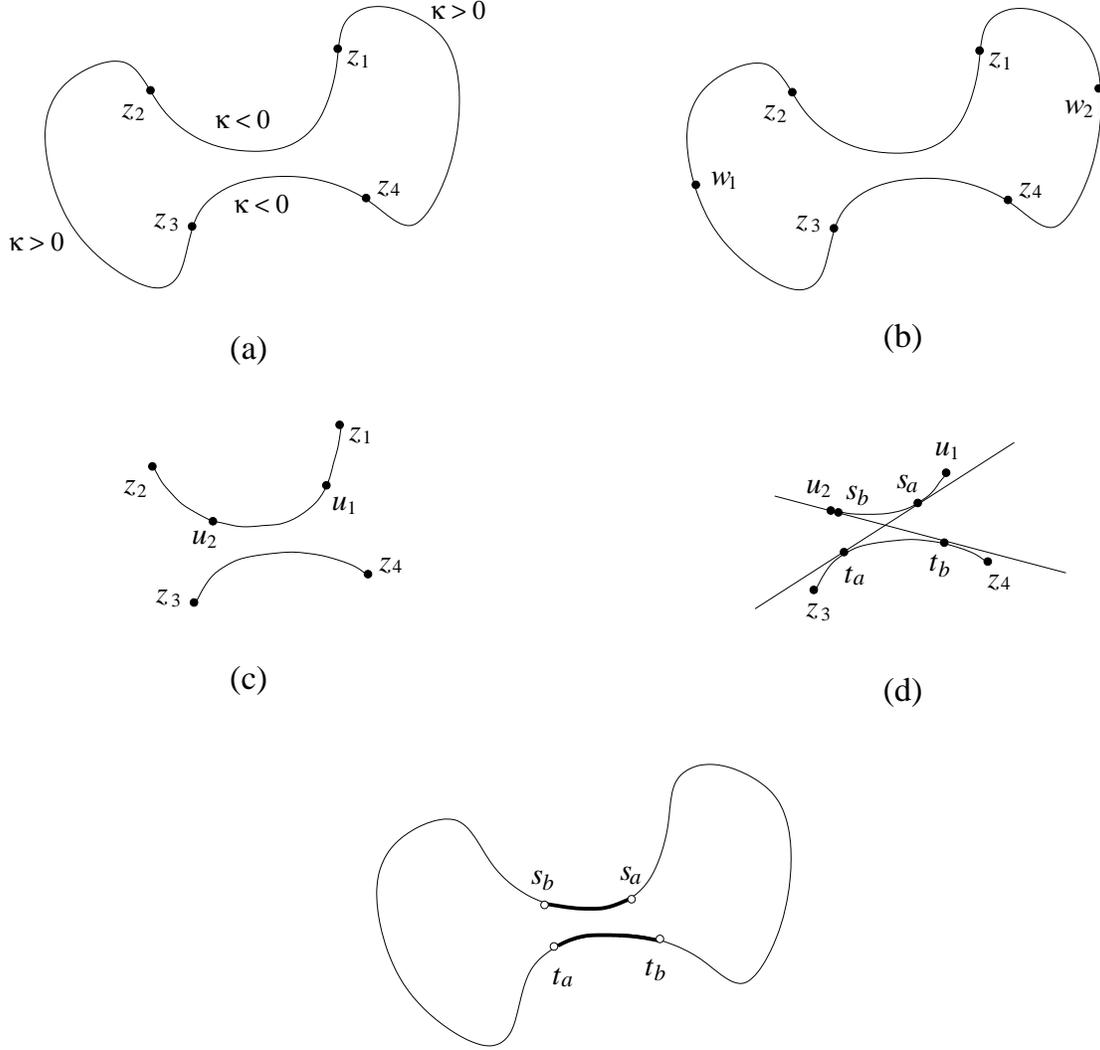
A call to the subroutine `March` finds the first pair of antipodal points  $s^*$  and  $t^*$ , if exists, in the direction of marching. No antipodal point exists on the interval  $(s_a, s^*)$  or  $(s^*, s_b)$ , which has been marched over. Shorten the unmarched interval of  $\mathcal{S}$  to  $(s^* + \epsilon, s_b)$  or  $(s_a, s^* - \epsilon)$ , and the unmarched interval of  $\mathcal{T}$  correspondingly. The antipodal angle changes sign at  $s^*$  but not over the interval discarded as a result of marching and shortening. Also  $\theta(s_a)$  and  $\theta(s_b)$  have the same sign. So we infer that the antipodal angles at the endpoints of the shortened interval must have different signs. By induction, the recursive call to the subroutine `Bisect` on line 7 or 8 will find the remaining  $k - 1$  pairs of antipodal points.  $\square$

Now let us conduct a time analysis of the procedure `Antipodal-Points`. Suppose there exist  $k$  pairs of antipodal points on the segments  $\mathcal{S}$  and  $\mathcal{T}$ . Again suppose  $\omega$  (binary) bits of precision is required. The subroutines `Bisect` and `March` each takes time  $O(\omega^2)$ , excluding the recursive calls within themselves. The total number of recursive calls is  $\Theta(k)$ . In case of no antipodal points, only the subroutine `March` is invoked once. Therefore the running time of `Antipodal-Points` is  $O((k + 1)\omega^2)$ .

## 5 Curve Preprocessing

In this section, we describe how to generate pairs of segments on  $\alpha$  that satisfy conditions (i)–(v) in Sections 2 and 3.1. The preprocessing consists of four consecutive steps overviewed below and illustrated in Figure 21.

1. Compute all points of *simple inflection* on  $\alpha$  and split the curve at these points. A point  $s$  is simple inflection if the curvature  $\kappa(s) = 0$  but  $\kappa'(s) \neq 0$ . In Figure 21(a), there are four



**Figure 21:** Four preprocessing steps. The two segments over  $(s_a, s_b)$  and  $(t_a, t_b)$  form one of the pairs that satisfy conditions (i)–(v) and will be searched for antipodal points by the algorithm in Section 4.

simple inflections  $z_1, z_2, z_3,$  and  $z_4$ . They divide  $\alpha$  into four segments, each of which satisfies  $\kappa > 0$  everywhere or  $\kappa < 0$  everywhere in its interior.

2. Split every segment with total curvature beyond  $[-\pi, \pi]$ . In Figure 21(b), the segments over  $[z_2, z_3]$  and  $[z_4, z_1]$  are split at the points  $w_1$  and  $w_2$ , respectively.
3. Enumerate all pairs of the resulting segments. For each pair shorten one or both segments if necessary until their endpoints have opposite normals. Consider the pair defined on  $[z_1, z_2]$  and  $[z_3, z_4]$  in Figure 21(c), for example. The segment on  $[z_1, z_2]$  is replaced with the segment  $[u_1, u_2]$  where  $N(u_1) = -N(z_3)$  and  $N(u_2) = -N(z_4)$ . After this operation, every pair will satisfy conditions (i)–(iii) in Section 2.
4. For each segment pair, determine if any of its endpoints with opposite normals are antipodal.

If yes, report the antipodal points and shorten the two segments slightly. This has established condition (iv) in Section 3. But condition (v) is not necessarily satisfied. We need to extract the sections on the two segments determined by the incident points of their common tangents. For example, in Figure 21(d), condition (v) is violated at  $u_1$  and  $z_3$  and at  $u_2$  and  $z_4$ . The two common tangents are incident on  $s_a, s_b, t_a,$  and  $t_b$ . Only the two sections  $(s_a, s_b)$  and  $(t_a, t_b)$  from the segment pair satisfy condition (v).

Preprocessing steps 1, 2, and 4 will be detailed in Sections 5.1, 5.2, and 5.3, respectively.

## 5.1 Points of Simple Inflection

Intuitively, a point of simple inflection is where the closed curve  $\alpha$  changes from convex to concave or from concave to convex. In general, a point  $s$  with  $\kappa(s) = \kappa'(s) = \dots = \kappa^{(k-1)}(s) = 0$  and  $\kappa^{(k)}(s) \neq 0$  is an *inflection point of order  $k$* . A second order inflection will not alter the convexity or concavity of its neighborhood and thus can be neglected for our purpose. Inflections of order higher than two are very rare and are also neglected in this paper.

Locating inflection points amounts to finding the zeros of the nonlinear function  $\kappa(s)$ . A global strategy such as the one combining linear search and backtracking with Newton's method would be complicated to program and yet cannot guarantee to find all zeros. Since the curve domain  $[0, D]$  is one-dimensional, we here employ a straightforward strategy that almost always finds all inflection points on  $\alpha$ . Divide the curve domain, say  $[0, D]$ , into  $m$  subintervals. Every subinterval whose endpoints have curvatures of different signs contains at least one point of simple inflection. We can then find one such inflection using bisection. This method will find all points of simple inflection provided that no two of them are closer than  $\frac{D}{m}$  apart. We need to choose  $m$  large enough so as not to miss an inflection point but not too large to make the computation inexpensive.

When discretizing the domain into  $m$  subintervals, we can also compute the total curvatures of the resulting  $m$  segments using the closed form (27) in Appendix C.<sup>18</sup> This information is then used for evaluating the total curvatures of segments partitioned by inflection points.

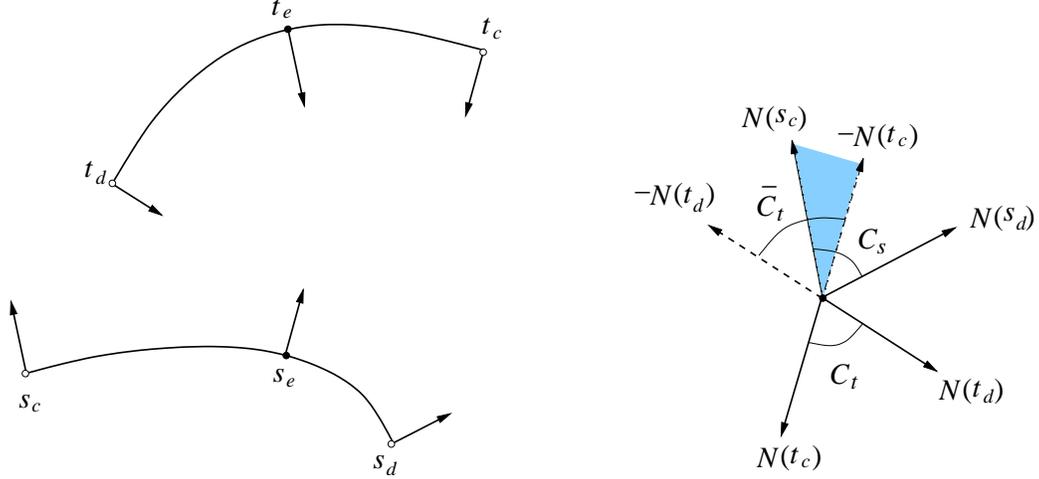
## 5.2 Opposing Normals

We enumerate all possible pairs of segments due to inflections and after splitting. How to preprocess one pair defined on, say  $[s_c, s_d]$  and  $[t_c, t_d]$ , so that their new endpoints will be opposite to each other? Let  $\mathcal{C}_s$  be the cone containing all the inward normals of the curve  $\alpha$  defined on  $[s_c, s_d]$  and  $\mathcal{C}_t$  be the cone containing all the inward normals defined on  $[t_c, t_d]$ . Let the cone  $\bar{\mathcal{C}}_t$  consist of all the outward normals on  $[t_c, t_d]$ . Next, intersect the two cones  $\mathcal{C}_s$  and  $\bar{\mathcal{C}}_t$ . The points whose inward normals (or outward normals) are the edges of the intersection cone bound exactly the portions of the segments that satisfy condition (ii). The remaining portions are discarded. If the cones  $\mathcal{C}_s$  and  $\bar{\mathcal{C}}_t$  do not intersect, the pair of segments will not contain any antipodal points.

Figure 22 illustrates the operation. In this example, the normals  $N(s_c)$  and  $-N(t_c)$  define the intersection cone. Using the procedure `point-with-normal`, we find  $s_e$  and  $t_e$  such that  $N(t_e) = -N(s_c)$  and  $N(s_e) = -N(t_c)$ . Clearly, antipodal points, if exist, can only appear on the curve segments over  $[s_c, s_e]$  and  $[t_e, t_c]$ .

---

<sup>18</sup>Here we assume that  $m$  is large enough such that none of the  $m$  segments has total curvature greater than (or less than)  $2\pi$ .



**Figure 22:** Intersecting the two cones of curve normals over  $[s_c, s_d]$  and  $[t_c, t_d]$ , respectively. Antipodal points may only appear on the two sections over  $[s_c, s_e]$  and  $[t_e, t_c]$ .

Relabel the endpoints of the extracted sections as  $s_c, t_c, s_d,$  and  $t_d$  such that  $s_c < s_d$ ,  $N(s_c) = -N(t_c)$ , and  $N(s_d) = -N(t_d)$ . We detect if  $s_c$  and  $t_c$  are antipodal or  $s_d$  and  $t_d$  are antipodal. If so, simply remove from each segment a very small portion containing the antipodal point and update the endpoints accordingly to maintain condition (ii). Let us refer to the segment over the updated open subdomain  $(s_c, s_d)$  as  $\mathcal{S}$  and the segment over the updated open subdomain  $(t_c, t_d)$  as  $\mathcal{T}$ .

### 5.3 Extracting Opposite Segments

By now  $\mathcal{S}$  and  $\mathcal{T}$  satisfy conditions (i)–(iv). Condition (v) states that  $N(s) \cdot \mathbf{r}(s) > 0$  for all  $s$  on  $\mathcal{S}$ . This is not necessarily satisfied. To further process the segments, we observe that the dot product  $N(s) \cdot \mathbf{r}(s)$  changes its sign where

$$N(s) \cdot \mathbf{r}(s) = 0;$$

that is, when the line passing through  $s$  and its opposite point  $t$  is a *common tangent* of  $\mathcal{S}$  and  $\mathcal{T}$ . The algorithm in Section 2.4 then finds all common tangents of  $\mathcal{S}$  and  $\mathcal{T}$ .

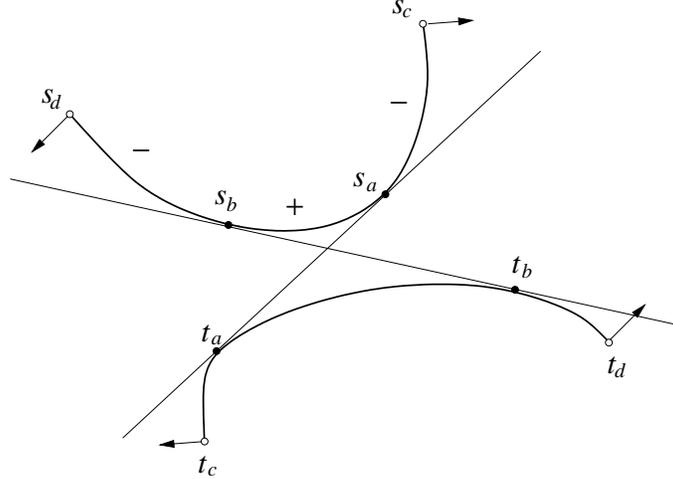
In Section 2.1 we showed that at most two common tangents exist on the segments. If no common tangent exists on  $\mathcal{S}$  and  $\mathcal{T}$ , the dot product  $N(s) \cdot \mathbf{r}(s)$  does not change its sign. We need only evaluate it at one point, say, the endpoint  $s_c$ . If the value is positive, then condition (v) is satisfied by  $\mathcal{S}$  and  $\mathcal{T}$ . Otherwise, the pair cannot contain any antipodal points and is not considered further.

Suppose that  $\mathcal{S}$  and  $\mathcal{T}$  have at least one common tangent. How do we extract the sections of  $\mathcal{S}$  and  $\mathcal{T}$  that satisfy condition (v) in general? In the case of one common tangent, denote by  $s_a$  and  $t_a$  the points of tangency on  $\mathcal{S}$  and  $\mathcal{T}$ , respectively. In the case of two common tangents, denote by  $s_b$  and  $t_b$  the second pair of tangency points such that  $s_a < s_b$ . The antipodal angle  $\theta(s_a)$  at  $s_a$  is either  $\frac{\pi}{2}$  when  $N(s_a) \times \mathbf{r}(s_a) > 0$  or  $-\frac{\pi}{2}$  when  $N(s_a) \times \mathbf{r}(s_a) < 0$ . Equation (10) simplifies to  $\theta'(s_a) = -\kappa(s_a)$ . There are four possibilities:

1.  $\theta(s_a) = \frac{\pi}{2}$  and  $\theta'(s_a) < 0$ . In this case,  $(s_a, s_b)$  and  $(t_a, t_b)$ , or  $(s_a, s_d)$  and  $(t_a, t_d)$  if only one common tangent exists, satisfies (v).

2.  $\theta(s_a) = \frac{\pi}{2}$  and  $\theta'(s_a) > 0$ . In this case,  $(s_c, s_a)$  and  $(t_c, t_a)$  satisfy condition (v). When the second common tangent exists,  $(s_b, s_d)$  and  $(t_b, t_d)$  also satisfy condition (v).
3.  $\theta(s_a) = -\frac{\pi}{2}$  and  $\theta'(s_a) < 0$ . Same as Case 2.
4.  $\theta(s_a) = -\frac{\pi}{2}$  and  $\theta'(s_a) > 0$ . Same as Case 1.

An example is shown in Figure 23. The two points of tangency  $s_a$  and  $s_b$  ( $t_a$  and  $t_b$ , respectively)



**Figure 23:** Extracting two concave segments over  $(s_a, s_b)$  and  $(t_a, t_b)$  which satisfy conditions (i)–(iv). The antipodal angle  $\theta$  is zero at the points of tangency  $s_a$  and  $s_b$  and alternates its sign (as labelled) on the three segments over  $(s_c, s_a)$ ,  $(s_a, s_b)$ , and  $(s_b, s_d)$ .

divide  $\mathcal{S}$  ( $\mathcal{T}$ , respectively) into three sections. Here  $\theta(s_a) = -\frac{\pi}{2}$  and  $\theta'(s_a) > 0$ . The dot product  $N(s) \cdot \mathbf{r}(s)$  is positive over  $(s_a, s_b)$  but negative over  $(s_c, s_a)$  and  $(s_b, s_d)$ . Clearly, antipodal points could only appear on the two sections over  $(s_a, s_b)$  and  $(t_a, t_b)$ , respectively.

## 6 Time Complexity

Let  $n$  be the number of inflection points and  $m$  the number of pairs of antipodal points on the curve  $\alpha$ . Suppose  $\omega$  bits are required for the precision.

A monotone segment between two points of simple inflection generally does not take the shape of a spiral with total curvature many times of  $2\pi$ . So we assume that  $\Theta(n)$  segments with total curvature in  $[-\pi, \pi]$  are generated by the dissection. There are  $O(n^2)$  pairs of segments satisfying conditions (i)–(v) introduced in Sections 2 and 3.1. The dominating cost of generating each pair after the dissection lies in the common tangent construction, which is  $O(\omega \log \omega)$  as analyzed in Section 2.3. The cost of preprocessing is also affected by the computation of inflection points, which can be approximated as  $O(n\omega)$  provided that the curve domain is discretized into  $\Theta(n)$  subintervals. Empirically we have found that the computation is still dominated by common tangent construction. So it is reasonable to approximate the total cost of preprocessing as  $O(n^2\omega \log \omega)$ .

There are a total of  $O(n^2 + m)$  calls to the procedures **Bisect** and **March** in Section 4. As given in Section 4.3, every such call requires  $O(\omega^2)$  low-level numerical steps. Therefore the time spent on finding all antipodal points is  $O((n^2 + m)\omega^2)$ , which remains to be the running time even when

preprocessing is taken into account. In practice, the number  $\omega$  of digits required for the precision is a constant. Then the running time becomes  $O(n^2 + m)$ . The bound is tight for some curves such as those in the family defined by (8), where  $m = \Theta(n^2)$  can be reached.

## 7 Arbitrary-Speed Curves

Until now we have assumed that the curve  $\alpha$  is unit-speed. In fact, all the described numerical procedures are still applicable when  $\alpha$  is arbitrary-speed. The convergence results in Sections 2.3, 4.2.1, and 4.2.2, for finding common tangents and antipodal points are clearly independent of parameterization. Now let us show that the local convergence rates will carry over as well.

Suppose  $\alpha(s)$  is not unit-speed. Let  $\sigma$  be the modified version of any of the previously described iteration functions  $h, f, g$  given in Sections 2.3, 4.2.1, and 4.2.2, respectively, such that  $s_{i+1} = \sigma(s_i)$ . There exists a unit-speed reparameterization  $\tilde{\alpha}(\tilde{s})$  of  $\alpha$  such that  $\|\tilde{\alpha}'(\tilde{s})\| = 1$  [27, p. 51]. From  $\alpha(s) = \tilde{\alpha}(\tilde{s}(s))$  we obtain the derivative

$$\alpha'(s) = \tilde{\alpha}'(\tilde{s}) \tilde{s}'(s).$$

Since  $\|\tilde{\alpha}'\| = 1$ , the derivative of the new parameter is

$$\tilde{s}'(s) = \|\alpha'(s)\|.$$

Denote by  $\tilde{\sigma}$  the corresponding iteration function for the unit-speed parameterization that we have studied before. So  $\tilde{\sigma}$  is one of  $f, g$ , and  $h$ . It follows that

$$\begin{aligned} \tilde{s}(s_{i+1}) &= \tilde{s}_{i+1} \\ &= \tilde{\sigma}(\tilde{s}_i) \\ &= \tilde{\sigma}(\tilde{s}(s_i)). \end{aligned}$$

Differentiating the above equation yields

$$\tilde{s}'(s_{i+1})\sigma'(s_i) = \tilde{\sigma}'(\tilde{s}(s_i)) \tilde{s}'(s_i). \quad (21)$$

The iteration functions under the two parameterizations are thus related by

$$\begin{aligned} \sigma'(s_i) &= \tilde{\sigma}'(\tilde{s}(s_i)) \tilde{s}'(s_i) / \tilde{s}'(s_{i+1}) \\ &= \tilde{\sigma}'(\tilde{s}(s_i)) \|\alpha'(s_i)\| / \|\alpha'(s_{i+1})\|. \end{aligned}$$

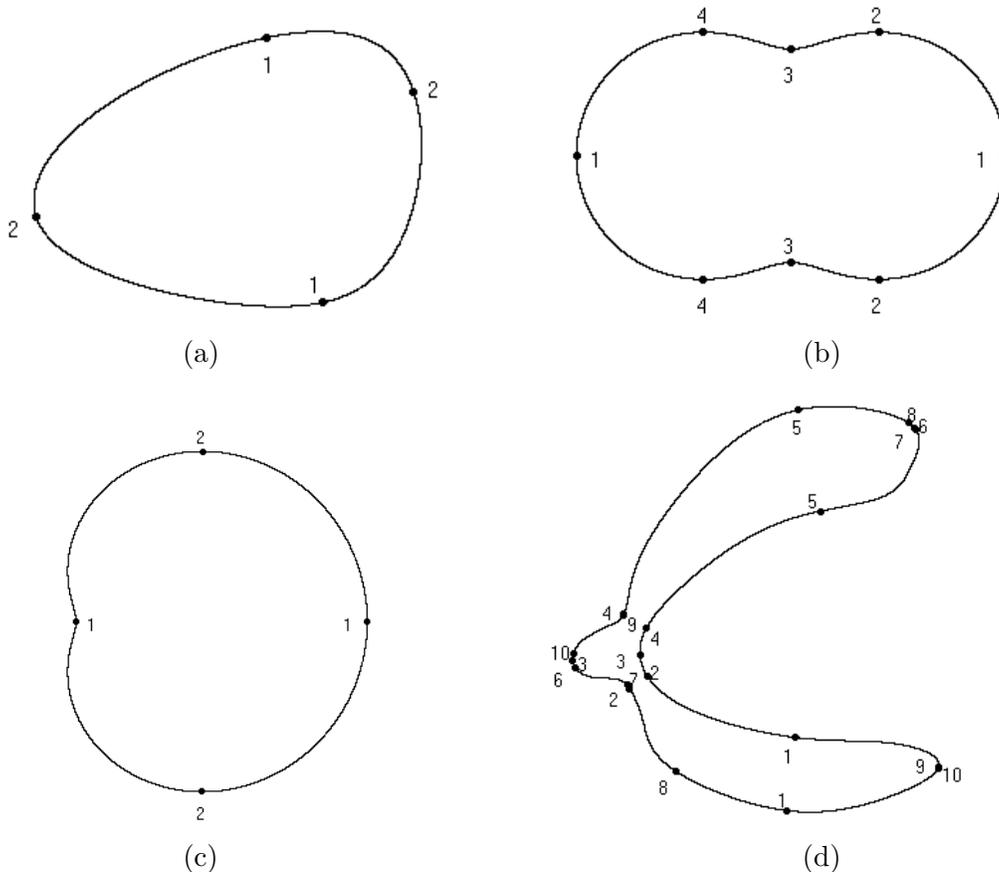
At an antipodal point  $s^*$  the above equation simplifies to

$$\begin{aligned} \sigma'(s^*) &= \tilde{\sigma}'(\tilde{s}^*) \frac{\|\alpha'(s^*)\|}{\|\alpha'(s^*)\|} \\ &= \tilde{\sigma}'(\tilde{s}^*). \end{aligned}$$

In the case  $\sigma'(s^*) = 0$ , where  $\sigma$  corresponds to the iteration function  $h$  in Section 2.3, we differentiate (21) and derive that

$$\sigma''(s^*) = \tilde{\sigma}''(\tilde{s}^*) \cdot \|\alpha'(s^*)\|.$$

Therefore for arbitrary-speed curves, the convergence rates of all three iterative procedures are the same as for the unit-speed curves. In other words, these rates are *independent of the parameterization*.



**Figure 24:** Antipodal points on four different shapes: (a) a convex cubic spline; (b) an elliptic lemniscate; (c) a limaçon; and (d) a non-convex cubic spline.

## 8 Experiments

We have implemented the algorithm for computing antipodal points in MS Visual C++ for arbitrary-speed curves. Appendix A offers details on the implementation of the primitives `point-with-normal`, `point-on-ray`, and `point-of-tangency`.

Figure 24 displays all antipodal points found on four different shapes: (a) a convex cubic spline; (b) an elliptic lemniscate described by  $\rho = \sqrt{6^2 \cos^2 \phi + 3^2 \sin^2 \phi}$  in polar coordinates; (c) a limaçon  $\rho = 4 + \frac{5}{2} \cos \phi$ ; and (d) a non-convex cubic spline. The first three examples each took about 10 milliseconds on a DELL Dimension PC with Pentium III 933 MHz CPU. The fourth example took 20 milliseconds. A non-degenerate closed convex curve, like the cubic spline in (a), has two pairs of antipodal points. We already gave a formal argument of the fact in the introduction. That four pairs of antipodal points exist on an elliptic lemniscate can be verified. A limaçon has two pairs of antipodal points at 0 and  $\pi$  and at  $\frac{\pi}{2}$  and  $\frac{3\pi}{2}$ , respectively.

The non-convex cubic spline in (d) has four of the five types of antipodal points classified in Figure 15. The two pairs of antipodal points labeled 6 and 10 are of Type A. Pair 8 is of Type B. Pairs 1, 3, 5, 7, and 9 are of Type D. And pairs 2 and 4 are of Type E.

## 9 Discussion

We have presented two algorithms that compute geometric substructures related to a simple, closed, and twice continuously differentiable curve. These structures include common tangents and antipodal points. Inflection points divide the curve into segments that are either convex everywhere or concave everywhere. Such monotonicity allows the design of various marching strategies with linear or quadratic convergence rates. To find all antipodal points, two such strategies are interleaved with bisection recursively based on the local geometry, which is determined from the signs of antipodal angles. The idea of curve dissection followed by the coupling of marching with bisection is potentially applicable to other computational problems involving curves.

Nonlinear programming may not be very effective at solving geometric problems, especially those with low dimensions. For instance, a nonlinear programming solution based on equations (7) to find antipodal points, inherently local, would have to rely heavily on initial guesses. It would be slow (when multiple guesses are used) and not guarantee to find any antipodal points, not to mention all of them. Our results demonstrate that computational efficiency and (almost) completeness can be achieved by exploiting both global and differential geometry.

The common tangent algorithm may be used as a subroutine to support more complex curve computation. For instance, in the author's recent work [17] it was employed to design a group of algorithms that construct the convex hulls of closed plane curves and polygons.

The described work is expected to be implemented with an Adept Cobra 600 robot as part of the ongoing research on localizing and grasping curved objects [13, 14]. We will need to develop a quality measure [25] that selects the best pair of antipodal points to place two fingers at.

In the following we would like to discuss in more details numerical issues, completeness, extensions, and analysis surrounding the presented curve computation scheme.

### 9.1 Completeness and Numerical Issues

The completeness of the presented algorithms is subject to locating all inflection points in the pre-processing. At present, the curve domain is partitioned into many subintervals on which bisection is invoked separately with the curvature function. In case multiple inflection points fall within the same subinterval, bisection will fail to find all of them, if any at all. Let  $v_{\max} = \max_t \|\alpha'(t)\|$  be the maximum speed of the curve  $\alpha$ . And let  $d_1$  be the minimum arc length between any two inflections on the curve. Discretization and bisection together will find all inflection points provided that each subinterval has size less than  $\frac{d_1}{v_{\max}}$ .

Another place where completeness may be compromised is when a pair of found antipodal points is stepped over in the search for more pairs. This situation will not happen provided that the step size is less than  $\frac{d_2}{v_{\max}}$ , where  $d_2$  is the minimum arc length between two adjacent pairs.

For special curves it may be possible to obtain lower bounds for  $d_1$  and  $d_2$  and an upper bound for  $v_{\max}$ . For general curves it seems rather unrealistic to estimate the spacing between inflection points and antipodal points before they are even found.

Locating inflection points may become a bottleneck. The strategy discussed above is inefficient and could dominate the running times of the given algorithms when the resolution used is too fine. The difficulty comes down to finding the roots of the one-variable nonlinear equation  $\kappa' = 0$ . Existing numerical methods such as bisection, Newton's, secant, etc. all sacrifice completeness. Although some bounds and methods for inflections on algebraic curves have been developed, much investigation is needed for parametric curves.

## 9.2 Extensions

The common tangent and antipodal point algorithms can be extended in a straightforward way to a curve that is piecewise twice continuously differentiable. The curve does not even need to be simple, that is, it can have self-crossings. Extension of the algorithms to a curved surface in 3D will likely hinge on how efficiently the surface can be partitioned into patches on which a search or an objective function assumes similar monotonicity.

Algebraic curves can be parametrized locally. So the iterative marching procedures that form the core of the two algorithms are expected to have their algebraic counterparts. Such an extension may be similar to the derivation of the curvature formula for algebraic curves from the one for parametric curves.

## 9.3 Computational Analysis

Our design technique based on dissection suggests a measure of the “combinatorial size” of a curve by the number of its inflections. Both algorithms for constructing common tangents and for finding antipodal points employ specialized numerical primitives (described in Appendix A) to complete a basic operation such as moving a point on the curve or determining if certain geometric condition holds. Such a routine call usually completes in tens to hundreds of iteration steps, whereas in discrete geometry a movement from one vertex or edge to another can be performed in one single step. Thus it seems reasonable to analyze the asymptotic running time of a curve processing algorithm in terms of the number of calls to numerical primitives.

## A Numerical Primitives

Numerical primitives are employed in the algorithms described in Sections 2 and 4. Each primitive operates on a monotone segment of  $\alpha$  over  $(a, b)$  with total curvature  $\Phi(a, b) \in [-\pi, \pi]$ .

The primitive `point-with-normal` $(a, b, \mathbf{n})$  finds a point  $c \in (a, b)$  on the curve  $\alpha$  such that the normal  $N(c)$  is in the direction of the vector  $\mathbf{n}$ . Thus  $c$  exists if and only if  $N(a) \times \mathbf{n}$  and  $N(b) \times \mathbf{n}$  have different signs. If so, we use the Newton-Raphson method to find  $c$  as the unique zero of the function  $\alpha'(s) \cdot \mathbf{n}$ . The iteration is given by

$$s_{k+1} = s_k - \frac{\alpha'(s_k) \cdot \mathbf{n}}{\alpha''(s_k) \cdot \mathbf{n}}.$$

In case Newton-Raphson fails, bisection guarantees to find  $c$ .

The primitive `point-of-tangency` $(a, b, c)$  finds a point  $e \in (a, b)$  at which a line through  $c$  is tangent to the curve  $\alpha$ . In order to use the procedure, such a point of tangency must uniquely exist. This is guaranteed when the primitive is called inside the procedure `Common-Tangent` described in 2.3. The point of tangency  $e$  is the unique root of the function  $(\alpha(s) - \alpha(c)) \times \alpha'(s)$  on the segment over  $(a, b)$ . We can again employ the Newton-Raphson method with the following iteration:

$$s_{k+1} = s_k - \frac{(\alpha(s_k) - \alpha(c)) \times \alpha'(s_k)}{(\alpha(s_k) - \alpha(c)) \times \alpha''(s_k)}.$$

When called inside `Common-Tangent`, the above function always has different signs at  $a$  and  $b$ . In case Newton-Raphson fails, bisection is used.<sup>19</sup>

The primitive `point-on-ray( $a, b, c$ )` is used by the marching procedure defined by (19) and illustrated in Figure 20. It determines if the curve segment over  $(a, b)$  intersects the ray of normal at  $c$  on  $\alpha$ , and finds the intersection if so. We know from Section 4.2.2 that such an intersection exists if and only if  $\alpha(a)$  and  $\alpha(b)$  are on different sides of the ray. The intersection, when exists, is the unique zero of the function  $(\alpha(s) - \alpha(c)) \cdot \alpha'(c)$  on  $\mathcal{S}$ . Similarly, we employ the Newton-Raphson method with iteration

$$s_{k+1} = s_k - \frac{(\alpha(s_k) - \alpha(c)) \cdot \alpha'(c)}{\alpha'(s_k) \cdot \alpha'(c)}.$$

Again, bisection is used as a backup.

Each of the three primitives is dominated by a Newton-Raphson call backed up by bisection. The Newton-Raphson method has quadratic local convergence rate as compared to bisection's linear convergence rate. In the worst case, Newton-Raphson fails and bisection has to be invoked. Every step of bisection halves the interval, adding one extra binary bit of precision to the solution. Suppose we require  $\omega$  binary places of precision. This is guaranteed by bisection in  $\omega + \log(b - a)$  iterations. The Newton-Raphson method, when converges, requires significantly fewer number of steps to achieve the same precision. So we may first employ Newton-Raphson for a fraction of  $\omega + \log(b - a)$  steps, and if it is not converging, resort to bisection next.

Thus  $O(\omega)$  is the number of numerical iterations carried out by any of the above primitive operations. For a fixed precision,  $\omega$  is treated as a constant.

## B Curve Tangents from an Exterior Point

Consider a point  $\mathbf{p}$  and a regular curve  $\alpha$  over the domain  $(a, b)$ , where  $\mathbf{p}$  does not lie on  $\alpha$ . How many tangent lines of  $\alpha$  pass through  $\mathbf{p}$ ? Here we investigate the case where  $\alpha$  is monotone and has total curvature  $\Phi(a, b) \in [-\pi, \pi]$ . The tangent count is needed in Section 2.2 for the classification of common tangent configurations.

Let  $L_a$  and  $L_b$  be the limits of tangent lines of  $\alpha(s)$  as  $s$  approaches  $a$  and  $b$ , respectively. If they intersect, let  $O$  be the point of intersection. This is shown in Figure 25. When  $\mathbf{p}$  is on  $L_a$  or  $L_b$ , a tangent line of  $\alpha$  passes through  $\mathbf{p}$  if and only if  $\mathbf{p}$  does not coincide with  $O$ ,  $a$ , or  $b$ .

The lines  $L_a$  and  $L_b$  together with  $\alpha$  partition the plane into five open regions. Every point in the region  $R_0$  lies on two different tangent lines. Every point in the region  $R_1$  or  $R_2$  does not lie on any tangent line to the curve.<sup>20</sup> Every point in the region  $R_3$  or  $R_4$  lies on exactly one tangent line. Below we describe a method to determine which region contains  $\mathbf{p}$ .

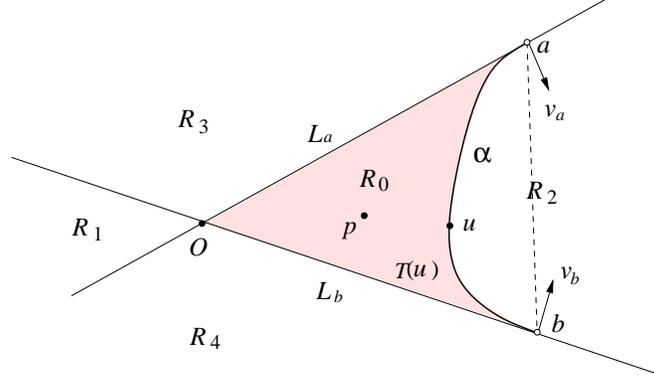
If  $\alpha$  is convex, we let  $\mathbf{v}_a = N(a)$  and  $\mathbf{v}_b = N(b)$ . Otherwise, we let  $\mathbf{v}_a = -N(a)$  and  $\mathbf{v}_b = -N(b)$ . These vectors  $\mathbf{v}_a$  and  $\mathbf{v}_b$  indicate the bending directions of  $\alpha$  at the points  $a$  and  $b$ , respectively. We observe that

$$\mathbf{p} \in R_0 \cup R_2 \quad \text{iff} \quad (\mathbf{p} - \alpha(a)) \cdot \mathbf{v}_a > 0 \quad \text{and} \quad (\mathbf{p} - \alpha(b)) \cdot \mathbf{v}_b > 0;$$

---

<sup>19</sup>A byproduct of the on-line convex hull algorithm in [30] is the  $O(\log n)$ -time construction of the two supporting lines from an external point to a convex polygon. This construction, a discrete version as compared to `point-of-tangency`, marches over vertices of the polygon quickly to the point of tangency based on a classification of the vertices relative to the point.

<sup>20</sup>In case  $L_a \parallel L_b$ , the region  $R_1$  disappears.



**Figure 25:** Five regions divided by a curve  $\alpha$  over  $(a, b)$  and the two tangent lines at  $a$  and  $b$ . The regions  $R_1$  and  $R_2$  include all points not on any tangent line to the curve. The regions  $R_3$  and  $R_4$  include all points on one tangent line. The region  $R_0$  includes all points on two tangent lines.

$$\begin{aligned}
 \mathbf{p} \in R_1 & \text{ iff } (\mathbf{p} - \alpha(a)) \cdot \mathbf{v}_a < 0 \text{ and } (\mathbf{p} - \alpha(b)) \cdot \mathbf{v}_b < 0; \\
 \mathbf{p} \in R_3 & \text{ iff } (\mathbf{p} - \alpha(a)) \cdot \mathbf{v}_a < 0 \text{ and } (\mathbf{p} - \alpha(b)) \cdot \mathbf{v}_b > 0; \\
 \mathbf{p} \in R_4 & \text{ iff } (\mathbf{p} - \alpha(a)) \cdot \mathbf{v}_a > 0 \text{ and } (\mathbf{p} - \alpha(b)) \cdot \mathbf{v}_b < 0.
 \end{aligned}$$

When  $\mathbf{p} \in R_0 \cup R_2$ , we still need to determine which of the two regions contains  $\mathbf{p}$ . First, we quickly check if  $\mathbf{p} \in \Delta aOb$ . If not, then  $\mathbf{p} \in R_2$ . If yes, we compute the point  $u$  on the curve which is the root of the function

$$(\mathbf{p} - \alpha(s)) \cdot \alpha'(s). \quad (22)$$

Newton-Raphson can be used with backup from bisection since the function has different signs at  $a$  and  $b$ . Clearly,  $\mathbf{p} \in R_0$  if  $(\mathbf{p} - \alpha(u)) \cdot \kappa(u)N(u) < 0$  and  $\mathbf{p} \in R_2$  if  $(\mathbf{p} - \alpha(u)) \cdot \kappa(u)N(u) > 0$ .

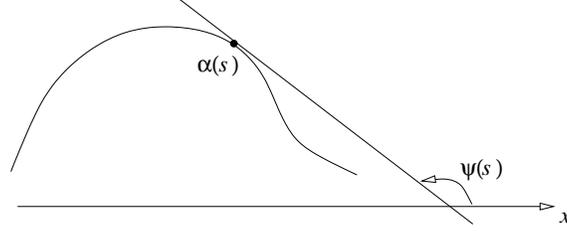
The worst case in computation happens when  $p$  lies in  $R_0$ . It involves finding the root of the function defined in (22) to determine that  $p$  lies in the region. For a precision of  $\omega$  binary digits, the root finding requires  $O(\omega)$  iteration steps the same as required by a primitive in Appendix A. So determining the number of tangent line(s) from a point takes  $O(\omega)$  steps.

## C Tangential Angle and Curvature

Let  $\alpha(s)$  be a regular plane curve defined over some interval with no self-crossings. The *speed* of the curve  $\|\alpha'(s)\| \neq 0$  everywhere.

The *tangential angle*  $\psi(s)$  is formed by the tangent  $\alpha'(s)$  with the  $x$  axis, as shown below. Choose a reference point  $s_0$  and introduce the arc length parameter  $\tilde{s} = \int_{s_0}^s \|\alpha'(u)\| du$ . Hence  $d\tilde{s}/ds = \|\alpha'(s)\|$  and  $\|d\alpha/d\tilde{s}\| = 1$ . The *curvature*  $\kappa(s)$  of  $\alpha$  is the rate of change of the tangential angle  $\psi(s)$  with respect to  $\tilde{s}$  [36, p. 134], that is,

$$\begin{aligned}
 \kappa &= \frac{d\psi}{d\tilde{s}} \\
 &= \frac{d\psi}{ds} \bigg/ \frac{d\tilde{s}}{ds}
 \end{aligned}$$



**Figure 26:** Tangential angle.

$$= \frac{\psi'(s)}{\|\alpha'(s)\|}. \quad (23)$$

In another definition [27, p. 57], the curvature measures the turning of the unit tangent vector  $T(s) = \alpha'(s)/\|\alpha'(s)\|$  along the unit normal direction  $N(s)$  where  $T(s) \times N(s) = 1$ . The Frenet formulas are known to hold:

$$\begin{aligned} T' &= \kappa \|\alpha'\| N; \\ N' &= -\kappa \|\alpha'\| T. \end{aligned} \quad (24)$$

We can easily derive one of the curvature definitions (23) and (24) from the other. For instance, if we start with definition (24), then

$$\begin{aligned} \kappa &= (T' \cdot N) / \|\alpha'\| \\ &= \left( \frac{dT}{ds} \cdot N \right) \frac{ds}{d\tilde{s}} \\ &= \frac{dT \cdot N}{d\tilde{s}} \\ &= \frac{d\psi}{d\tilde{s}}. \end{aligned} \quad (25)$$

Let  $\alpha(s) = (x(s), y(s))$ . Then

$$\begin{aligned} T &= (x', y') / \|\alpha'(s)\| = (x', y') / \sqrt{x'^2 + y'^2}, \\ N &= (-y', x') / \sqrt{x'^2 + y'^2}. \end{aligned}$$

Substituting these terms into (25) yields a formula for evaluating the curvature:

$$\begin{aligned} \kappa &= \left( \frac{(x'', y'')}{\sqrt{x'^2 + y'^2}} + \frac{d}{ds} \left( \frac{1}{\sqrt{x'^2 + y'^2}} \right) (x', y') \right) \cdot \frac{(-y', x')}{\sqrt{x'^2 + y'^2}} \Big/ \sqrt{x'^2 + y'^2} \\ &= \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{\frac{3}{2}}} \end{aligned}$$

If  $\kappa \neq 0$ , the *center of curvature* lies along the direction of  $\kappa N(s)$  at distance  $\frac{1}{|\kappa|}$  from the point  $\alpha(s)$ . The *osculating circle*, when  $\kappa \neq 0$ , is the circle at the center of curvature with radius  $\frac{1}{|\kappa|}$ ,

which is called the *radius of curvature*. The osculating circle approximates the curve locally up to the second order.

The *total curvature* of  $\alpha$  over a closed interval  $[a, b]$  measures the rotation of the unit tangent  $T(s)$  as  $s$  changes from  $a$  to  $b$ :

$$\begin{aligned}\Phi(a, b) &= \int_a^b \kappa \|\alpha'(s)\| ds \\ &= \psi(b) - \psi(a).\end{aligned}\tag{26}$$

If  $\alpha$  is closed over  $[a, b]$ , then  $\Phi(a, b) = 2\pi$  (the Fenchel equality [39, p. 60]). If the total curvature over  $[a, b]$  is within  $(0, 2\pi]$ , it has a closed form:

$$\Phi(a, b) = \begin{cases} \arccos(T(a) \cdot T(b)), & \text{if } T(a) \times T(b) \geq 0; \\ 2\pi - \arccos(T(a) \cdot T(b)), & \text{otherwise.} \end{cases}\tag{27}$$

When the tangent makes several full revolutions<sup>21</sup> as  $s$  increases from  $a$  to  $b$ , it cannot be determined just from  $T(a)$  and  $T(b)$ .

## Acknowledgement

Support for this research was provided in part by Iowa State University through a Faculty Development Grant, and in part by the National Science Foundation through a CAREER award IIS-0133681. The author would like to thank David Persky for helping with the implementation, David Kriegman for pointing out the possibility of globally solving low-dimensional problems without resorting to optimization, and Ravi Janardan for suggesting to analyze the presented algorithms and for relating to the common tangent algorithm for two convex polygons by Preparata and Hong [31]. Thanks to the anonymous reviewers for providing constructive feedback that helped improve the structure and presentation of the paper. Portions of this paper were presented at the 2002 IEEE International Conference on Robotics and Automation (ICRA '02) [16] and the Fifth International Workshop on Algorithmic Foundations of Robotics (WAFR '02) [15]. The author is also grateful to the anonymous reviewers of these two papers for their valuable feedback and comments.

---

<sup>21</sup>For example, the curve is a spiral.

## References

- [1] S. Bespamyatnikh. An efficient algorithm for the three-dimensional diameter problem. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 137–146, 1998.
- [2] Antonio Bicchi. Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity. *IEEE Transactions on Robotics and Automation*, 16(6):652–662, 2000.
- [3] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: a review. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 348–353, 2000.
- [4] Andrew Blake and Michael Taylor. Planning planar grasps of smooth contours. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 834–839, 1993.
- [5] Randy C. Brost and Kenneth Y. Goldberg. A complete algorithm for synthesizing modular fixtures for polygonal parts. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 535–542, 1994.
- [6] B. Chazelle, H. Edelsbrunner, L. J. Guibas, and M. Sharir. Diameter, width, closest line pair and parametric searching. *Discrete & Computational Geometry*, 10:183–196, 1993.
- [7] I-Ming Chen and Joel W. Burdick. Finding antipodal point grasps on irregularly shaped objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2278–2283, 1992.
- [8] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, ii. *Discrete & Computational Geometry*, 4:387–421, 1989.
- [9] P. Erdős. On sets of distances of  $n$  points in Euclidean space. *Magy. Tud. Akad. Mat. Kut. Int. Kozi*, 5:165–169, 1960.
- [10] T. N. T. Goodman. Inflections on curves in two and three dimensions. *Computer Aided Geometric Design*, 8(1):37–50, 1991.
- [11] B. Grunbaum. A proof of Vazsonyi’s conjecture. *Bulletin of Research Council Israel*, 6(A):77–78, 1956.
- [12] J. Hong, G. Lafferriere, B. Mishra, and X. Tan. Fine manipulation with multifinger hands. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1568–1573, 1990.
- [13] Yan-Bin Jia. Grasping curved objects through rolling. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 377–382, 2000.
- [14] Yan-Bin Jia. Localization on curved objects using tactile information. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 701–706, 2001.
- [15] Yan-Bin Jia. Computation on parametric curves with applications in localization and grasping. In *Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics*, 2002.

- [16] Yan-Bin Jia. Curvature-based computation of antipodal grasps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1571–1577, 2002.
- [17] Yan-Bin Jia. On the convex hulls of parametric plane curves. Submitted to *Computational Geometry: Theory and Applications*, <http://www.cs.iastate.edu/~jia/papers/convex-hull.pdf>, 2003.
- [18] Zexiang Li and S. Shankar Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal of Robotics and Automation*, 4(1):32–44, 1988.
- [19] Dinesh Manocha and John F. Canny. Detecting cusps and inflection points in curves. *Computer Aided Geometric Design*, 9(1):1–24, 1992.
- [20] Xanthippi Markenscoff, Luqun Ni, and Christos H. Papadimitriou. The geometry of grasping. *International Journal of Robotics Research*, 9(1):61–74, 1990.
- [21] Xanthippi Markenscoff and Christos H. Papadimitriou. Optimum grip of a polygon. *International Journal of Robotics Research*, 8(2):17–29, 1989.
- [22] Brian Mirtich and John Canny. Easily computable optimum grasps in 2-D and 3-D. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 739–741, 1994.
- [23] B. Mishra, J. T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(4):541–558, 1987.
- [24] Bud Mishra. Workholding: analysis and planning. In *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 53–57, 1991.
- [25] Bud Mishra. Grasp metrics: optimality and complexity. In Ken Goldberg et al., editor, *Algorithmic Foundations of Robotics*, pages 137–165. A. K. Peters, Boston, MA, 1995. Proceedings of the 1994 Workshop on the Algorithmic Foundations of Robotics.
- [26] Van-Duc Nguyen. Constructing force-closure grasps. *International Journal of Robotics Research*, 7(3):3–16, 1988.
- [27] Barrett O’Neill. *Elementary Differential Geometry*. Academic Press, Inc., 1966.
- [28] Jean Ponce, Darrell Stam, and Bernard Faverjon. On computing two-finger force-closure grasps of curved 2D objects. *International Journal of Robotics Research*, 12(3):263–273, 1993.
- [29] Jean Ponce, Steven Sullivan, Jean-Daniel Boissonnat, and Jean-Pierre Merlet. On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 821–827, 1993.
- [30] F. P. Preparata. An optimal real time algorithm for planar convex hulls. *Communications of the ACM*, 22:402–405, 1979.
- [31] F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, 20(2):87–93, 1977.

- [32] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry: an Introduction*. Springer-Verlag, 1985.
- [33] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Inc., 1988.
- [34] Edgar A. Ramos. Construction of 1-d lower envelopes and applications. In *Proceedings of the ACM Symposium on Computational Geometry*, pages 57–66, 1997.
- [35] Anil Rao and Ken Goldberg. Manipulating algebraic parts in the plane. *IEEE Transactions on Robotics and Automation*, 11(4):598–602, 1995.
- [36] John W. Rutter. *Geometry of Curves*. Chapman & Hall/Crc, Boca Raton, Florida, 2000.
- [37] M. Sakai. Inflection points and singularities on planar rational cubic curve segments. *Computer Aided Geometric Design*, 16:149–156, 1999.
- [38] J. Matoušek and O. Schwarzkopf. A deterministic algorithm for the three-dimensional diameter problem. *Computational Geometry: Theory and Applications*, 6:253–362, 1996.
- [39] Eugene V. Shikin. *Handbook and Atlas of Curves*. CRC Press, Inc., 1995.
- [40] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, 2nd edition, 1993.
- [41] M. Teichmann. *Grasping and Fixturing: a Geometric Study and an Implementation*. PhD thesis, New York University, 1995.
- [42] Marek Teichmann and Bud Mishra. Reactive robotics I: reactive grasping with a modified gripper and multifingered hands. *International Journal of Robotics Research*, 19(7):697–708, 2000.
- [43] Jeffrey C. Trinkle. A quantitative test for form closure grasps. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1670–1677, 1992.
- [44] A. C.-C. Yao. On constructing minimum spanning trees in  $k$ -dimensional space and related problems. *SIAM Journal of Computing*, 11(4):721–736, 1982.