Complete Subdivision Algorithms, I: Intersection of Bezier Curves

Chee K. Yap*
Department of Computer Science
Courant Institute, New York University
New York, NY 10012, USA
and
School of Computer Science and Engineering
Seoul National University
yap@cs.nyu.edu

August 10, 2005

Abstract

We give the the first complete subdivision algorithm for the intersection of two Bezier curves F, G which finitely many common points, possibly with tangential intersections. Our new approach to robust subdivision algorithms is based on geometric separation bounds, and on a new criterion for detecting non-crossing intersection of curves. Our algorithm is fully adaptive, being based only on bigfloat number computations. In particular, we avoids manipulation of algebraic numbers as well as resultant computations. It is designed to be competitive with current algorithms on "nice" inputs.

Dedicated to Godfried Toussaint on the occasion his 60th Birthday.

1 Introduction

The intersection and analysis of algebraic curves and surfaces is a fundamental problem in many areas of geometric modeling [11]. Many practical algorithms are based on free-form curves and surfaces [6, 5]. In this paper, we consider Bezier curves. All current algorithms are inexact, leading to well-known nonrobustness problems. Let us look at a fundamental reason for this.

A Bezier curve F is represented by a sequence $P(F) = (p_0, \ldots, p_m)$ of control points. Let $CH(F)$ denote the convex hull of $p_0, \ldots, p_m$. A pair (F, G) of

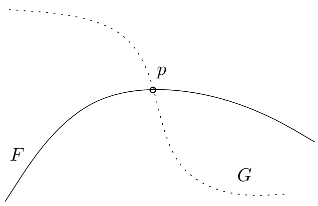
*The work is supported in part by NSF/ITR Grant #CCR-0082056, and was conducted at Seoul National University.

Bezier curves is called a **candidate pair** if $CH(F) \cap CH(G)$ is non-empty. The standard algorithmic paradigm for intersecting Bezier curves is based on two ideas. First, using the Bezier curve property that F is contained in $CH(F)$, the algorithm can restrict attention to candidate pairs only. Second, the principle algorithmic operation is **subdivision**: this operation divides a curve F into two pieces F_0, F_1 using De Casteljau's algorithm. The generic intersection algorithm goes as follows:

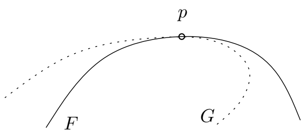
GENERIC INTERSECT (F, G)
Input: F, G are Bezier curves with no common component
Output: A list L of pairs (F_i, G_i), $i \geq 1$, each indicating an "intersection"
1. if $CH(F) \cap CH(G) = \emptyset$, return.
2. if the diameters of $CH(F)$ and $CH(G)$ are both $< \varepsilon$, append (F, G) to L and return.
3. Subdivide the curve, say F, with the larger diameter into subcurves F_0, F_1.
4. (Recursively) Call INTERSECT(F_0, G) and INTERSECT(F_1, G).

This algorithm depends on a constant $\varepsilon > 0$. For purposes of display, such a constant is justifiable. But for topological analysis of curve arrangements, we want output pairs (F', G') to represent a *unique* intersection. But the above algorithm might output a pair (F', G') that is wrong in one of two ways: it may have *no* intersection or have *multiple* intersections.

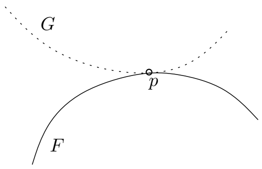
Let p be an intersection point of F and G. Under the **standard assumption** that F, G have no common component, then p is an isolated point of $F \cap G$. The intersection at p can be **tangential** or **transversal**, depending on whether the tangents of F and G at p are coincident or not. Alternatively, we can classify p as a **crossing** or **non-crossing** intersection, depending on whether the curves cross each other at p or not. Non-crossing intersections must be tangential, and transversal intersections must be crossing. The 3 possibilities are illustrated in Figure 1.



(a) Tansversal, crossing



(b) Tangential, crossing



(c) Tangential, non-crossing

Figure 1: Intersections: (a) Transversal (b) Tangential

Can the use of ε be avoided? It seems plausible that if F and G have only crossing intersections, then we can design an intersection algorithm based on subdivision that does not use any ε-cutoff. In fact, this is true, and is implied by our intersection algorithm. In any case, the key issue is how to detect non-crossing intersections. Recently, Wolpert [17, 16] addressed this issue for nonsingular algebraic curves. But the class of curves she addresses and the ones addressed in this paper are not directly comparable – Bezier curves form a rather special subclass of algebraic curves, but Bezier curves may be singular. She introduced the technique of generalized Jacobi curves to detect non-crossing intersection of non-singular curves. Like us, she using subdivision of space and deliberately avoids the manipulation of algebraic numbers (unlike the traditional approach based on cylindrical decomposition). But her approach still uses algebraic techniquess such as resultants and root isolation. Such algebraic techniques are[1] not considered adaptive, as they have a fixed overhead cost. A more recent paper of Seidel and Wolper [14] addresses computing the topological arrangement of plane algebraic curves; again, a combination of subdivision and algebrac methods are used. In contrast, the only algebraic information we use are algebraic zero bounds. Otherwise, we perform purely numerical computations using bigfloat numbers and primitive geometric operations such as computing convex hulls and intersecting curves with a line. It is not at all obvious that we can achieve our exact curve intersection goals using only geometric and numerical approximations. For instance, a purely adaptive/numerical version in the Wolpert-Seidel setting is not known.

Current intersection algorithms deploy a variety of criteria for detecting intersection points. These are typically **partial criteria**: either a **rejection criterion** that detects no intersection or an **acceptance criterion** that affirms an intersection. A **complete criterion** is one that is both an acceptance and rejection criterion. The most well-known rejection criterion is the **convex hull criterion**: if F, G intersects then their convex hull must intersect. Thus, the generic intersection algorithm above uses only this criterion. Sederberg and Meyers [13] gave an acceptance criterion based on hodographs which affirm the presense of a transversal intersection. However there are no known acceptance criterion for non-crossing intersections; we will provide one in Section XXX.

Overview of Our Approach. 1. The fundamental motivation of this work is to design a fully adaptive subdivision algorithm for Bezier intersection. The qualification "fully adaptive" for subdivision algorithms is necessary to distinguish our approach from approaches which use a combination of algebraic methods with numerical methods, as in [16, 17, 14]. Our algorithm uses numerical and geometric approximations; the only algebraic information we exploit is the existence of separation bounds. This requirement forces us to design various new subroutines. Another aspect of adaptivity means that these bounds are only invoked in the worst case scenario. For "nice inputs", the running time is expected to be much faster than implied by the worst case bound.

2. The main analytical tool we introduce is geometric separation bounds. They answer such questions as: *How close can two curves be, if they do not*

[1] Nicola Wolpert and Raimund Seidel (personal communication and talk at Dagstuhl Workshop 2005), observed that their implementation does not necessarily run faster for easy examples. This is attributable to the non-adaptive aspects of their algorithm.

intersect? or *How close can a point q be to a curve if q is not on the curve?*. These bounds (denoted by various Δ's) provide stopping criteria for our numerical and subdivision procedures. The Δ-bounds depend only on the input data (degree and norms of the input curves and points). In particular, they do not change when we subdivide the curves.

The structure of our algorithm are oblivious to these bounds. This means that if any improved Δ-bounds are available, we can automatically incorporate them into our algorithm without any change in the algorithmic logic.

3. In Section 3, we develop the first complete criterion for detecting non-crossing intersection of **elementary Bezier Curves**. By an "elementary curve" we mean the graph of a convex or concave function. The non-crossing criterion has preconditions: so to apply this criterion, we need to design a **micro process** that creates the necessary preconditions for applying the criterion (Section XXX) and also adaptive methods for evaluating a certain "alpha function" (Section XXX).

4. What shall we do with non-elementary Bezier curves? A general Bezier curve have **critical points**, defined in such a way that their removal will ensure that the curve is elementary (Section XXX). There are algebraic techniques for breaking up a general algebraic curve at such critical points (e.g., [2, 10]). But this would involve algebraic, non-adaptive methods which we wish to avoid. Instead, we show how subdivisioning with separation bounds can completely isolate such critical points, and we provide subalgorithms for detecting such critical points. Combined with the above micro process, we obtain our final algorithm.

5. All our computational procedures are ultimately reduced to ring operations $(+, -, \times)$ on binary bigfloats (i.e., rational numbers of the form $n2^m$ where $n, m \in \mathbb{Z}$). These operations are carried out exactly. We avoid using algebraic or even general rational numbers for reasons of efficiency.

Points or Bezier curves whose parameters are given by bigfloats are called **direct objects**. At various places, we will need **indirect objects** where the parameters will generally involve algebraic numbers. For instance, intersecting a direct line with a direct Bezier curve will yield an indirect point. We must provide means of approximating indirect objects by direct objects – computing with indirect objects is necessarily iterative.

Related Work. The computational literature on algebraic curves and surfaces is very large and diverse. We may roughly divide the computational approaches into two distinct viewpoints: (A) The **Algebraic Viewpoint** treats curves and surfaces as systems of algebraic equations to be solved, usually using symbolic or algebraic techniques. Such "algebraic algorithms" are exact and complete. (B) The **Geometric Viewpoint** prefers curves and surfaces in parametric form, usually solved using numerical techniques such as homotopy or subdivision. Such "geometric algorithms" are often incomplete but are widely used in practice. The Algebraic Viewpoint is more general, with impressive advances made in the last 20 years (e.g., [3]). Nevertheless, many algebraic algorithms are not considered practical. The curves and surfaces in applications are usually bounded subsets ("patches") of an algebraic set. The geometric algorithms directly manipulate such patches; the algebraic algorithms treat complete algebraic sets, often assumed to be irreducible. This fact reduces the applicability of algebraic algorithms. To specify patches of an algebraic set, one could use semi-algbraic formulas (i.e., introduce inequalities). But it is not easy, say, to specify a particular branch of a curve in the neighborhood of a self-intersection using this method. The computation and topological analysis of real plane curves is a well studied problem [1, 12], but the worst case complexity is prohibitive: $O(n^{27}L^3)$ time for a curve $F(X, Y) = 0$ of degree n and L-bit coefficients. When algebraic algorithms are combined with numerical techniques, more practical algoritms can be achieved, as demonstrated by Hong [9]. Recently, computational geometers have begun to address curves and surfaces [4, 7, 15]. These papers provide efficient and complete algebraic algorithms for low degree surfaces and curves. More generally, the goals of these papers is to make algorithms under the algebraic viewpoint more efficient and complete, e.g., via careful considerations of the primitives and treatment of degeneracies, which theoretical papers may gloss over. In contrast, the goal of the present paper is to make algorithms under the geometric viewpoint completely robust while preserving their adaptive efficiency.

The work most directly comparable to ours is Wolpert and Seidel's [17, 16, 14], mentioned above. Their approach has features of the algebraic viewpoint (e.g., the use of resultant computation and root isolation), but her box-refinement technique is a subdivision approach commonly found under the geometric viewpoint. A recent topologically correct geometric algorithms that is based on purely subdivision scheme is [**?**]. The paper shows how to construct a polygonal surface that is isotopic to an given surface given in implicit form. However, the implicit surface must not have singularities – thus avoiding the kind of separation bounds introduced in our paper.

More generally, our approach might be described as **numerical algebraic computation** whose ostensible aim is to do algebraic computation without algebraic manipulations. We only do numerical approximations, but for exact decisions, we exploit algebraic zero bounds.

Organization of Paper.

- In Section 2, we provide several geometric separation bounds for algebraic curves.
- In Section 3, we upper bound the 2-norm of the implicit equation of a Bezier curve. This is necessary for applying the results of Section 2 to Bezier curves.
- In Section 4, we introduce elementary curves and give a "non-crossing intersection criterion" (NIC) for a pair of elementary curves.
- Section 5 shows how we can actually apply NIC for a particular micro-candidate pair.

- Section 6 gives the details for intersecting a curve with a line; this special case is a key subroutine in the general algorithm.
- Section 7 reduces the tangency criterion of Section 4 to the numerical evaluation of an expression to sufficient accuracy.
- Section 8 shows to reduce the detection of non-crossing intersection to an application of our tangency criterion.
- Section 9 addresses the problem of non-elementary curves: how to eliminate singular points, y-tangencies and inflexion points.
- Section 10 gives the overall intersection algorithm in terms of two stages: the traditional "macro stage" followed by a "micro stage".
- We conclude in Section 11 with future work and open problems.
- Two appendices with details are attached.

2 Geometric Separation Bounds for Algebraic Curves

Our key mathematical tool is the idea of separation bounds.

Suppose $A(x,y), B(x,y) \in \mathbb{Z}[x,y]$. Let $\|A\|_k$ denote the k-norm of the vector of the coefficients of A (we use $k = 1, 2$ and $k = \infty$). If $p \in \mathbb{R}^n$ is a point in Euclidean space, its k-norm has a similar notation, $\|p\|_k$. We may omit the subscript k when referring to the 2-norm, i.e., Euclidean distance. Consider the curves F and G defined by the equations $A = 0$ and $B = 0$, respectively. A pair (p, q) where $p \in F$ and $q \in G$ such that the line through p, q is normal to p at F and q at G is called an **(F,G)-antipodal pair**. In the following, we assume there are only finitely many pairs of (F,G) antipodal pairs. This implies that A, B are relatively prime (i.e., they have no common component).

We use a multivariate root bound of Yap [18] (Theorem 11.45): Let $\Sigma = \{A_1, \ldots, A_n\} \subseteq \mathbb{Z}[X_1, \ldots, X_n]$ be a system of n polynomials, not necessarily homogeneous. Suppose Σ has finitely[2] many complex zeros and $(\xi_1, \ldots, \xi_n) \in \mathbb{C}^n$ is one of these zeros. Assume $d_i = \deg(A_i)$ and let

$$K := \max\{\sqrt{n+1}, \|A_1\|_2, \ldots, \|A_n\|_2\}. \tag{1}$$

Fact 1 (Yap) *If $(\xi_1, \ldots, \xi_n)$ is a zero of Σ and $|\xi_i| \neq 0$, $i = 1, \ldots, n$, then*

$$|\xi_i| > (2^{3/2} N K)^{-D} 2^{-(n+1)d_1 \cdots d_n}$$

where

$$N := \binom{1 + \sum_{i=1}^n d_i}{n}, \qquad D := \left(1 + \sum_{i=1}^n \frac{1}{d_i}\right) \prod_{i=1}^n d_i. \tag{2}$$

[2]Note that this bound, unlike a similar bound from Canny, does not require that the system has finitely many roots at infinity.

Our first result gives a lower bound on the distance between an antipodal pair.

THEOREM 1 *Suppose F and G have finitely many antipodal pairs, and are defined by the equations $A = 0$ and $B = 0$ where*

$$m = \deg(A), \quad n = \deg(B), \qquad \|A\|_2 = a, \quad \|B\|_2 = b.$$

If (p, q) is a (F, G)-antipodal pair and $p \neq q$ then $\|p - q\| \geq \Delta_1(m, n, a, b)$ where

$$\begin{aligned} \Delta_1(m, n, a, b) &:= (2^{\frac{3}{2}} NK)^{-D}\ 2^{-12m^2n^2} \\ K &= \max\{\sqrt{13}, 4ma, 4nb\}, \\ N &= \binom{3 + 2m + 2n}{5}, \\ D &= m^2n^2(3 + \frac{4}{m} + \frac{4}{n}). \end{aligned}$$

Proof. By definition of antipodal pairs, $p = (p.x, p.y)$ and $q = (q.x, q.y)$ and the distance $h = \|p - q\|$ satisfy the following system of five-variate polynomial equations:

$$\begin{aligned} \mathcal{A}_1 &: A(p) = 0, \\ \mathcal{A}_2 &: B(q) = 0, \\ \mathcal{A}_3 &: h^2 - \|p - q\|^2 = 0 \\ \mathcal{A}_4 &: \left\langle (\frac{dA}{dy}(p),\ -\frac{dA}{dx}(p)), (p - q) \right\rangle = 0, \\ \mathcal{A}_5 &: \left\langle (\frac{dB}{dy}(q),\ -\frac{dB}{dx}(q)), (p - q) \right\rangle = 0, \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ denotes inner product. Since A, B has finitely many antipodal pairs, this system is zero dimensional. To apply Fact 1, note that $\|\mathcal{A}_1\|_2 = a$, $\|\mathcal{A}_2\|_2 = b$ and $\|\mathcal{A}_3\|_2 = \sqrt{13}$. Next, $\|A_y\|_2 \leq ma$ where $A_y = dA/dy$. Since $\|A + B\|_2 \leq \|A\|_2 + \|B\|_2$, we get $\|A_y(p) \cdot (p.x - q.x)\|_2 \leq 2ma$. Hence

$$\|\mathcal{A}_4\|_2 = \|A_y(p) \cdot (p.x - q.x) - A_x(q) \cdot (p.y - q.y)\|_2 \leq 4ma. \tag{3}$$

Similarly, $\|\mathcal{A}_5\|_2 \leq 4nb$. Also, $d_1 = d_3 = m, d_2 = d_4 = n, d_5 = 2$. The result follows by plugging into Fact 1. **Q.E.D.**

The following separation bound is simpler:

THEOREM 2 *Let the curves $A = 0$ and $B = 0$ be relatively prime. If p, q are distinct points in the intersection of these two curves then $\|p - q\| \geq \Delta_2(m, n, a, b)$ where*

$$\begin{aligned} \Delta_2(m, n, a, b) &:= (2^{\frac{3}{2}} NK)^{-D}\ 2^{-12m^2n^2} \\ K &= \max\{\sqrt{13}, m, n\}, \end{aligned}$$

$$
\begin{aligned}
N &= \binom{3+2m+2n}{5}, \\
D &= m^2n^2(3+\frac{4}{m}+\frac{4}{n}).
\end{aligned}
$$

Proof. Now p and q satisfy the following system of equations:

$$
\begin{aligned}
\mathcal{A}_1 &: A(p)=0, \\
\mathcal{A}_2 &: B(q)=0, \\
\mathcal{A}_3 &: h^2-\|p-q\|^2=0 \\
\mathcal{A}_6 &: A(q)=0, \\
\mathcal{A}_7 &: B(p)=0.
\end{aligned}
$$

This system is zero-dimensional because A, B are relatively prime. The application of Fact 1 is direct in this case. **Q.E.D.**

We next bound how close a point p can get to an algebraic curve $A(x,y)=0$, without actually being on it. By (L,ℓ)**-bit floating point numbers** (floats, for short) we mean numbers of the form $x = m2^{k-L}$ where $m, k \in \mathbb{Z}$ with $|m| < 2^L$ and $0 \le k \le \ell$. Note that $-L \le \lg|x| < \ell$ (where lg is $\log_2$). We simply say "L-bit floats" for (L,L)-bit floats.

THEOREM 3 *Let $q=(u,v)$ be a point whose coordinates are (L,ℓ)-bit floats, $\ell \ge 2$, and $A(u,v) \ne 0$. If the curve $A=0$ does not contain a circle centered at q, and p is a point on the curve $A=0$ then*

$$\|p-q\| \ge \Delta_3(m,a,L,\ell)$$

where

$$
\begin{aligned}
\Delta_3(m,a,L,\ell) &:= (2^{\frac{3}{2}}NK)^{-D}\ 2^{-8m^2} \\
K &= 2^{L+\ell+1}\max\{2^L, ma)\}, \\
N &= \binom{3+2m}{3}, \\
D &= m^2(3+\frac{4}{m}).
\end{aligned}
$$

Proof. We may assume p is a closest point on A to q and $h=\|p-q\|$. Thus three variables $p.x, p.y, h$ satisfy the system,

$$
\begin{aligned}
\mathcal{A}_1 &: A(p)=0, \\
\mathcal{A}_3 &: h^2-\|p-q\|^2=0, \\
\mathcal{A}_4 &: \left\langle (\frac{dA}{dy}(p),\ -\frac{dA}{dx}(p)), (p-q) \right\rangle = 0.
\end{aligned}
$$

This system is zero-dimensional since the curve $A=0$ does contain a circle about q. We have $\|\mathcal{A}_1\| = a$, $\|\mathcal{A}_3\| \le \sqrt{1+2(2^{2\ell}+2^{\ell+1}+1)} < 2^{1+\ell}$ (assuming $\ell \ge 2$).

For $\|\mathcal{A}_4\|$, we have $\|A_x(p) \cdot q.x\| < 2^\ell ma$ but $\|A_x(p)\| = \|A_x(p) \cdot p.x\| \le ma$ (because $p.x$ is a variable). Hence $\|A_x(p) \cdot (p.x - q.x)\| \le ma(1 + 2^\ell)$. But instead of $\|\mathcal{A}_3\|$ and $\|\mathcal{A}_4\|$, we use the integer polynomials $2^{2L}\mathcal{A}_3$ and $2^L\mathcal{A}_4$, with $\|2^{2L}\mathcal{A}_3\| < 2^{2L+\ell+1}$ and $\|2^L\mathcal{A}_4\| \le ma(1+2^\ell)2^L < ma2^{1+\ell+L}$. The result follows by plugging into Fact 1. **Q.E.D.**

This result is used, for instance, in the Beneath-Beyond test at the end of Section 5(6?). The requirement that $A = 0$ does not contain a circle centered at q does not affect our application to Bezier curves, because circular arcs are non-Bezier.

The Δ-Separation Property. Note that $\Delta_1(m, n, a, b) < \Delta_2(m, n, a, b)$. A pair (p, q) is said to be **critical** for (F, G) if it is an (F, G)-antipodal pair or if they both belong to $F \cap G$. For any $\Delta > 0$, we say F, G have the **Δ-separation property** if every critical pair (p, q) of (F, G) with $p \ne q$ satisfies $\|p - q\| > \Delta$. In the following, we study the interaction of pairs of curves that satisfies this separation property. The **diameter** of any set $S \subseteq \mathbb{R}^2$ is defined as

$$\mathrm{diam}(S) := \sup\{\|p - q\| : p, q \in S\}.$$

Thus, if F, G have the Δ-separation property and $\mathrm{diam}(F \cup G) \le \Delta$, then there can be no critical pair (p, q) with $p \ne q$.

Note that $\Delta > 0$ depends only on the initial input curves F, G; subsequent subdivision of these curves do not change the Δ.

REMARK: If F, G is a pair of parallel lines or concentric circles, then our assumption that there are finitely many antipodal pair fails. More generally, if F contains an offset of G, or vice-versa, there are infinitely many antipodal pairs. We conjecture[3] that the converse is also true. If the conjecture holds, then note that a separation property still holds (even though our proof breaks down).

3 Bounding the 2-Norm of Bezier Curves

Our previous bounds hold for general algebraic curves, but somewhat better bounds will now be shown for Bezier curves. But a preliminary step is to derive a 2-norm bound for Bezier curves.

Let F be a Bezier curve with control points $P(F) = (p_0, \ldots, p_m)$. Let $CH(X)$ denote the convex hull of a finite set X of points, i.e., the smallest convex set containing X. We shall write $CH(P)$ instead of the more correct "$CH(P(F))$". Call $CH(P)$ the **convex hull** of F, and denoted[4] $CH(F)$.

[3]Sung Woo Choi has recently shown that our conjecture holds. Indeed, his proof allows F and G to be real analytic.

[4]There is a chance for confusion, as one may reasonably imagine that "$CH(F)$" refers to the convex hull of the set of points in F. There is one place in this paper where we use this notion, and that will be pointed out.

In order to apply the separation bounds, we need to determine the algebraic equation of F, and to bound its norm. We have

$$\begin{aligned}
F(t) &= \sum_{i=0}^{m} p_i \binom{m}{i} t^i (1-t)^{m-i} \\
&= \sum_{i=0}^{m} p_i \binom{m}{i} t^i \left(\sum_{j=0}^{m-i} t^j (-1)^j \binom{m-i}{j} \right) \\
&= \sum_{i=0}^{m} \sum_{j=0}^{m-i} \binom{m}{i,j} t^{i+j} (-1)^j p_i \\
&= \sum_{k=0}^{m} t^k \sum_{\ell=0}^{k} p_\ell \binom{m}{\ell, m-k} (-1)^{k-\ell} \\
&= \sum_{k=0}^{m} t^k \left[\begin{array}{c} a_k \\ b_k \end{array} \right]
\end{aligned}$$

where the last equation serves to define the constants a_k, b_k. Note that we write $\binom{m}{i,j}$ for $\frac{m!}{i!j!(m-i-j)!}$. Writing $F(t) = (F_1(t), F_2(t))$, we have

$$F_1(t) = \sum_{k=0}^{m} a_k t^k, \qquad F_2(t) = \sum_{k=0}^{m} b_k t^k.$$

We bound the coefficients of $F_1(t)$ and $F_2(t)$.

Lemma 4 *Suppose the control points in $(p_0, \ldots, p_m)$ have (L, ℓ)-bit floats in their coordinates. Then $2^L a_k$ is an integer, and $|a_k| \le 2^{L+k}\binom{m}{k}$, for $k = 0, \ldots, m$.*

Proof. The fact that $2^L a_k$ is an integer follows from the definition of (L, ℓ)-bit floats.

$$\begin{aligned}
a_k &= \sum_{\ell=0}^{k} (-1)^{k-\ell} p_\ell . x \binom{m}{\ell, m-k} \\
|a_k| &\le 2^L \sum_{\ell=0}^{k} \binom{m}{\ell, m-k} \\
&= 2^L \binom{m}{k} \sum_{\ell=0}^{k} \binom{k}{\ell} \\
&= 2^{L+k} \binom{m}{k}. \qquad (4)
\end{aligned}$$

Q.E.D.

We get an identical bound for $|b_k|$.

2-Norm of the Implicit Equation. The implicit equation for $F(t)$ is given by the resultant

$$B(x,y) = \mathtt{res}_t(F_1(t)-x, F_2(t)-y).$$

Thus $B(x,y)$ is the determinant of the $2m \times 2m$ matrix:

$$M = \left[\begin{array}{ccccccc}
a_m & a_{m-1} & \cdots & a_1 & a_0 - x & \cdots & \\
 & a_m & \cdots & a_2 & a_1 & \cdots & \\
 & & \ddots & & & \cdots & \\
 & & & a_m & a_{m-1} & \cdots & a_0 - x \\
b_m & b_{m-1} & \cdots & b_1 & b_0 - y & & \\
 & b_m & \cdots & b_2 & b_1 & \cdots & \\
 & & \ddots & & & \cdots & \\
 & & & b_m & b_{m-1} & \cdots & b_0 - y
\end{array}\right] \tag{5}$$

Note that the constant term in the first (resp., last) m rows is $a_0 - x$ (resp., $b_0 - y$).

We now bound the 2-norm of this determinant. The main tool is a multivariate Hadamard-type bound from [18, Sect. 11.11]; the univariate version was introduced by Graham and Goldstein [8]. To apply this bound, view each entry of the matrix M in (5) as a polynomial in x and y, and replace it by an upper bound on its 1-norm. Using the bound in Lemma 4, this gives the matrix M_1:

$$M_1 = \left[\begin{array}{ccccccc}
2^{L+m}\binom{m}{0} & 2^{L+m-1}\binom{m}{1} & \cdots & 2^{L+1}m & 2^L+1 & \cdots & \\
 & 2^{L+m}\binom{m}{0} & \cdots & 2^{L+2}\binom{m}{2} & 2^{L+1}m & \cdots & \\
 & & \ddots & & & \cdots & \\
 & & & 2^{L+m}\binom{m}{0} & 2^{L+m-1}\binom{m}{1} & \cdots & 2^L+1 \\
2^{L+m}\binom{m}{0} & 2^{L+m-1}\binom{m}{1} & \cdots & 2^{L+1}m & 2^L+1 & & \\
 & 2^{L+m}\binom{m}{0} & \cdots & 2^{L+2}\binom{m}{2} & 2^{L+1}m & \cdots & \\
 & & \ddots & & & \cdots & \\
 & & & 2^{L+m}\binom{m}{0} & 2^{L+m-1}\binom{m}{1} & \cdots & 2^L+1
\end{array}\right] \tag{6}$$

The generalized Hadamard bound for multivariate polynomials says that $\|\det(M)\|_2$ is bounded by the 2-norm of each row of M_1. Alternatively, this bound says that

$$\|\det(M)\|_2 \le (\nu_1)^n (\nu_2)^m$$

where ν_1 is the $(2,1)$-norm of $F_1(t) - x$ and ν_2 is the $(2,1)$-norm of $F_2(t) - y$, relative to t. We define the "$(2,1)$-norm relative to x_1" of an n-variate polynomial $p(x_1, \ldots, x_n) = \sum_{i=0}^{k} c_i x_1^i$ to be

$$\sqrt{\sum_{1}^{k} \|c_i\|_1^2}$$

where c_i's are polynomials in $x_2, \ldots, x_n$. That is, we replace each c_i in p by its 1-norm, and then take the 2-norm of the resulting polynomial (which is a univariate polynomial in x_1).

Lemma 5
(i) The 1-norm of each row in M_1 is bounded by $1 + 2^L 3^m$.
(ii) The 2-norm of each row in M_1 is bounded by $2^L 3^m$.

Proof. (i) The 1-norm of each row in M_1 is equal to

$$1 + \sum_{k=0}^{m} 2^{L+k} \binom{m}{k} = 1 + 2^L 3^m. \tag{7}$$

(ii) The 2-norm of each row can be written as $\sqrt{x^2 + y^2}$ where $y = 1 + 2^L$ and x^2 is the sum of the squares of all the other entries in the row. We use the fact that

$$\sqrt{x^2 + y^2} \leq x + y - 1$$

if $x \geq y \geq 2$. Hence the 2-norm of a row is at most $x + y - 1$. But $x + y$ is bounded by the 1-norm, which is at most $1 + 2^L 3^m$. Hence the 2-norm is at most $2^L 3^m$. **Q.E.D.**

The main result is:

Theorem 6 *Let F be a Bezier curve of degree m, with control points which are L-bit floating point numbers. Then F satisfies an an integer polynomial equation $A(x, y) = 0$ of degree m where*

$$\|A\|_2 \leq (16^L 9^m)^m$$

Proof. The curve $F = (F_1, F_2)$ satisfies the equation $B(x, y) = 0$ where $B(x, y) = \det(M)$ where M is given in (5). The numbers in M are are (L, ℓ)-bit floats, so the matrix $2^L M$ has integer values. Let $A(x, y) = \det(2^L M) = 2^{2mL} B(x, y)$ is an integer matrix. The multivariate version of the Graham-Goldstein lemma [18, Sect. 11.11] says that $\|A\|_2$ is bounded by the product of the 2-norms of each row of the matrix $2^L M_1$ where M_1 is given by (6). As a consequence of Lemma 5, the 2-norm of each row of $2^L M_1$ is at most $2^{2L} 3^m$, Since there are $2m$ rows, the product is $(2^{2L} 3^m)^{2m}$, as stated in the theorem. **Q.E.D.**

Remark that once the equation of F is bounded as in this theorem, this bound applies to all subcurves of F obtained by subdivision of F.

4 Elementary Curves

This section introduces the notion of elementary curves and considers intersection between two such curves. The main result is a complete criterion for non-crossing intersection of elementary curves.

Let $C^1[a, b]$ denote the set of bounded, continuous differentiable real functions defined on the interval $[a, b]$ with $a < b$. If $f \in C^1[a, b]$, then its **graph** is the parametrized curve $\{F(t) : t \in [a, b]\}$ where $F(t) = (t, f(t))$. Let $\mathcal{G}[a, b]$ denote the set of all graphs defined by functions in $C^1[a, b]$. Also, $\mathcal{G}$ is the union of $\mathcal{G}[a, b]$, over all $[a, b]$. Unless otherwise indicated, we will assume $t \in [a, b]$ and $[a, b] = [0, 1]$. All our curves have finite length. We shall also write "$F = F[a, b]$" to indicate that $F \in \mathcal{G}[a, b]$.

Elementary Curves. The line segment connecting $F(0)$ and $F(1)$ is called the **base segment** of F. We call F is an **elementary curve** if the union of F and its base segment forms a simple close curve that bounds a convex region. If F is elementary and F lies *above* (resp., *below*) its base segment, we call it A**-elementary** (resp., B**-elementary**).

The unit normal to F at $F(t)$ is given by

$$\begin{aligned} u_F(t) &= \left(\frac{-f'(t)}{\sqrt{1+f'(t)^2}}, \frac{1}{\sqrt{1+f'(t)^2}}\right) && (8) \\ &= (\cos(\theta_F(t)), \sin(\theta_F(t))) && (9) \end{aligned}$$

where $\theta_F(t)$ is called the **normal angle** at $F(t)$. It follows from (8) that $\theta_F(t) \in (0, \pi)$ where $0 < t < 1$. For simplicity, we assume that $0 < \theta_F(t) < \pi$ even when $t = 0$ and 1. Elementary curves can be characterized as follows:

- F is A-elementary iff $\theta_F(t)$ is non-increasing.
- F is B-elementary iff $\theta_F(t)$ is non-decreasing.

Let $n_F(t)$ denote the **normal line** through $F(t)$ along the direction $u_F(t)$. Consider the family $\{n_F(t) : t \in [0, 1]\}$ of normal lines for an A-elementary curve $F \in \mathcal{G}[0, 1]$. Call $n_F(0)$ and $n_F(1)$ the **end normals**. Divide the normal line $n_F(t)$ into two half-lines: an upper part $a_F(t)$ and a lower part $b_F(t)$ with $F(t)$ as the common end point of the half-lines. The half-lines $a_F(t)$ sweep a region that is bounded by $a_F(0)$, F and $a_F(1)$: this is called the **upper swept region** and denoted $A(F)$. We regard $a_F(0)$ as a closed half-line so that the region $A(F)$ is a closed region (in particular, $F \subseteq A(F)$). See Figure 2. When we extend $a_F(0)$ and $a_F(1)$ until they meet, we obtain a cone $C(F)$ that contains $A(F)$.

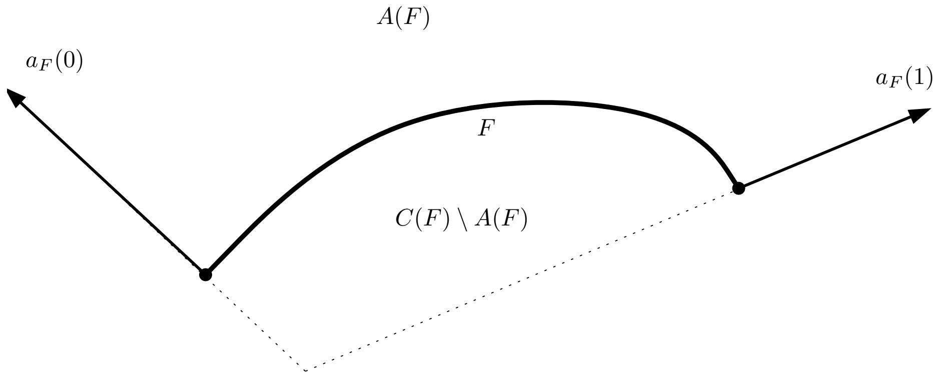


Figure 2: Upper swept region $A(F)$

LEMMA 7

(a) *The region $A(F)$ is stratified by the family of half-lines $\{a_F(t) : t \in [0,1]\}$, i.e., every point in $A(F)$ belongs to a unique half-line $a_F(t)$ and every half-line $a_F(t)$ is contained in $A(F)$.*

(b) *The cone $C(F)$ is swept by the family of normal lines $\{n_F(t) : t \in [0,1]\}$, i.e., every point in $C(F)$ belongs to at least one line $n_F(t)$.*

Proof.
(a) It is clear that $a_F(t) \subseteq A(F)$. To show that each $p \in A(F)$ belongs to a unique half-line, suppose $p \in a_F(t) \cap a_F(s)$ for some $t < s$. Since the vector $F(s) - F(t)$ has a positive x-component, and $\theta_F(s) > \theta_F(t)$, the normals $n_F(s)$ and $n_F(t)$ must intersect below the line through $F(s), F(t)$. Thus $p = b_F(s) \cap b_F(t)$, contradicting our assumption on p.
(b) In view of part (a), we only have to show that the region $C(F) \setminus A(F)$ is swept by the normals. This region is convex, and so it is divided by every half-line $b_F(t)$ into two convex parts, denoted $L(t)$ and $R(t)$. Let $L(t)$ denote the part that is left of $b_F(t)$ (this is the side to the left of the unit direction $u_F(t)$). We claim that for all $t \in [0,1]$, the region $L(t)$ is swept by the set of half-lines $\{b_F(t') : t' \in [0,t]\}$. The result is true for $t = 0$. If it is true for all $t' < t$ then we see that it is true for t, by a continuity argument. So the claim holds for all $t \in [0,1]$, by transfinite induction. The desired result follows, since $L(1) = C(F) \setminus A(F)$. **Q.E.D.**

Elementary Couples. Fix an A-elementary curve $F \in \mathcal{G}[0,1]$ as before. Suppose $G \in \mathcal{G}[a,b]$ is another elementary curve for some $a < b$. We call (F,G) an **elementary couple** if

(i) $G(a) \in a_F(0) \setminus \{F(0)\}$ and $G(b) \in a_F(1) \setminus \{F(1)\}$.

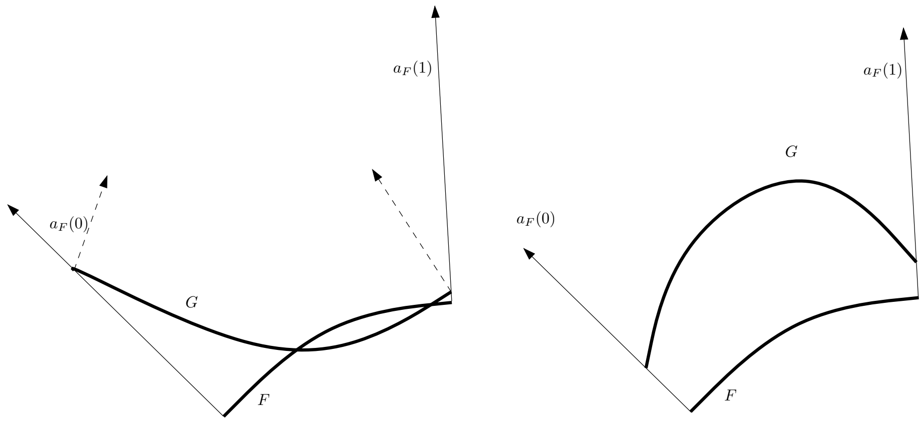
(ii) The entire curve G lies inside the cone $C(F)$.

See Figure 3. Note that G can be an A- or B-elementary; accordingly, we call (F,G) an AA- or an AB-couple.

Condition (i) says that the base segment of G connects the two rays $a_F(0)$ and $a_F(1)$. Moreover, the base segments of F and G are required to be disjoint. Note that condition (ii) allows the curve G to go outside the upper swept region $A(F)$ – this happens iff F and G intersects. See illustration in Figure 3.

In general, we need to treat another possibility: where (F,G) forms a BA**- or** BB**-couple**. But by symmetric, we focus only on AB and AB-couples.

Let $p \in F$ and $q \in G$. Call (p,q) a (F,G)**-closest pair** if $p \in F$ and $q \in G$ and $\|p - q\| = \min\{\|p' - q'\| : p' \in F, q' \in G\}$. If L is the normal line to F at p, then any point in $G \cap L$ is called a G**-projection** of point p. In general, a point p may have no G-projection or have multiple G-projections. Note that the pair (p,q) is an (F,G)-antipodal pair iff p is a F-projection of q, and q is a G-projection of p. In particular, if F and G intersect tangentially at p, then (p,p) is an (F,G)-antipodal pair.



(a) (b)

Figure 3: Elementary Couple (F, G): two cases

LEMMA 8 *Let $(F, G) \in \mathcal{G}[0, 1] \times \mathcal{G}[a, b]$ be a couple. If $G \subseteq A(F)$ then there is a unique continuous function $s : [0, 1] \to [a, b]$ such that each point $G(s(t))$ is a G-projection of $F(t)$.*

Proof. Since G lies in the upper swept region $A(F)$, Lemma 7(a) ensures that for all $t \in [0, 1]$, $n_F(t)$ intersects G. Our lemma amounts to saying that $n_F(t)$ intersects G exactly once. Clearly $n_F(t)$ intersects G an odd number of times. However, since the region bounded by G and its base segment is convex, no line can intersect G more than twice. Hence G is intersected exactly once by $n_F(t)$, say at the point $G(s(t))$. The continuity of the function $s(t)$ then follows from the continuity of the parameterization of $n_F(t)$ and the continuity of the curve G. **Q.E.D.**

COROLLARY 9 *For any couple (F, G), if $G \subseteq A(F)$ then the angular function*

$$\alpha : [0, 1] \to (-\pi, \pi) \tag{10}$$

given by $\alpha(t) = \theta_F(t) - \theta_G(s(t))$ is well-defined and continuous. Here, $s(t)$ is the continuous function in Lemma 8.

If the dependence on F, G is to be indicated, we may write $\alpha_{F,G}$ and $s_{F,G}$ for these functions.

Complete Non-Crossing Intersection Criterion. In the following, let (F, G) be a couple. Thus the angular function α $(= \alpha_{F,G})$ and associated function s is defined. Our main result for testing non-crossing intersection is as follows:

Theorem 10 (Non-crossing Intersection Criterion (NIC)) *Let (F, G) be a couple. Suppose F, G have the Δ-separation property and the diameter of $F \cup G$ is less than Δ.*
(i) If $\alpha(0)\alpha(1) \leq 0$ then F and G intersect tangentially, in a unique point.
(ii) If $\alpha(0)\alpha(1) > 0$ then F and G are disjoint.

Proof. If $F \cap G$ contains two distinct points p and q, then $\|p-q\| \leq \mathrm{diam}(F \cup G) \leq \Delta$. But this contradicts the Δ-separation property. We conclude that $F \cap G$ is either empty or contains a unique point. Thus, if $F \cap G$ intersects, they intersect tangentially. This also shows that $G \subseteq A(F)$. Therefore, the functions $s : [0, 1] \to [a, b]$ and α in (10) are well-defined and continuous.
(i) Assume $\alpha(0)\alpha(1) \leq 0$. So there exists $0 < t < 1$ such that $\alpha(t) = 0$. The pair $(p, q) = (F(t), G(s(t)))$ is therefore antipodal. If $p \neq q$, the Δ-separation property implies $\|p - q\| > \Delta$. But this contradicts our assumption that $F \cup G$ has diameter $\leq \Delta$. Thus $p = q$, i.e., F and G intersect tangentially.
(ii) Suppose, by way of contradiction, that F and G intersect at some (necessarily tangential) point p. Write $p = F(t_0)$ for some t_0. Then $\alpha(t_0) = 0$. Since $\alpha(0)\alpha(1) > 0$, we may assume $\alpha(0) > 0$ and $\alpha(1) > 0$ (the other case, $\alpha(0) < 0$ and $\alpha(1) < 0$, is similar). Since p is a tangential intersection and F is assumed to be below G, we see that $\alpha(t_0^-) > 0$ and $\alpha(t_0^+) < 0$. Then, by continuity of the function $\alpha(t)$, there exists $t_1 \in (t_0, 1)$ such that $\alpha(t_1) = 0$. Thus $(F(t_1), G(s(t_1)))$ is an antipodal pair. Again, we argue that $F(t_1) = G(s(t_1))$ must be a tangential intersection of F and G. This contradicts the Δ-separation property. **Q.E.D.**

In the following sections, we need to address two issues: (1) how create the necessary preconditions for applying the NIC above, and (2) device an adaptive method to determine the signs of the angles $\alpha(0)$ and $\alpha(1)$ in the NIC.

5 Intersection With A Line

In this section, we give a complete algorithm for a special case of Bezier curve intersection: where one of the curves is a straight line. This special case will be a basic computation tool in the general algorithm.

The reader may use any standard reference for Bezier curves, e.g., [6]. Let F be a Bezier curve with control points $(p_0, \ldots, p_m)$. Call F an **elementary Bezier curve** if

- $p_0.x < p_1.x < \cdots < p_m.x$.
- Each point p_i is an extremal point of $CH(F)$ (i.e., there is a line ℓ through p_i such that all the other control points lies strictly to one side of ℓ.

The problem of intersecting an arbitrary Bezier curve with a line is treated in standard textbooks [6]. The line ℓ is usually taken as the x-axis. But complete subdivision algorithm was known even for this special case.

Bezier Parametrization versus Graph Parametrization. For any paramter $t \in [0,1]$, the standard de Casteljau algorithm computes a point $F(t)$ on the curve F. Moreover, $F(0) = p_0$ and $F(1) = p_m$. Thus F can be regarded as a parametric curve $F = F[0,1]$. When F is an elementary Bezier curve, we now have two distinct parametrizations of F: the Bezier parametrization with domain $[0,1]$ and the graph parametrization with domain $[a,b]$ where $a = p_0.x$ and $b = p_m.x$. We call these two parametrizations the **Bezier parametrization** and the **graph parametrization**, respectively. In any context, we will make it clear which paramtrization is used.

The domain of Bezier parametrizations need not be always $[0,1]$. For any $a < b$, the curve $F[a,b]$ in the Bezier parametrization is well defined. However, our usage will have the restriction $[a,b] \subseteq [0,1]$. This simply means that $F[a,b]$ is the subcurve of $F[0,1]$. This notation allows us to express a Bezier curve as a subcurve of some other curve $F[0,1]$. This representation is important because the values a, b might be indirect objects (in the sense of Section 1), and we want to relate $F[a,b]$ to a direct object. For instance, a Bezier curve $F = F[a,b]$ may be subdivided into two subcurves F_0, F_1 at the point $F(t)$, where $F_0 = F[a,t]$ and $F_1 = F[t,b]$.

In either Bezier or graph parametrizations, we write "$F = F[a,b]$" to indicate that the domain of F is $[a,b]$. For elementary curves, we always assume $F(t).x < F(t').x$ for $t < t'$. We can similarly speak of the normal line $n_F(t)$, the normal angle $\theta_F(t)$, etc, with respect to the parameterization.

Representation of Intersection Points. Note that if F intersects ℓ tangentially, then this must be a non-crossing intersection. Hence we may simply speak of two kinds of intersections only: tangential and transversal. A candidate pair (F, ℓ) is called an **intersection representation** (or "intersection rep" for short) in the following cases.

- (Exact Type) ℓ intersects exactly one endpoint of F but not the relative interior of the control hull $CH(F)$. See Figure 4(a) and (b).
- (Transversal Type) ℓ intersects the relative interior of a base segment of F. See Figure 4(c).
- (Tangential Type) ℓ does not intersect the base segment of F but intersects the relative interior of $CH(F)$. Moreover, the diameter of $CH(F)$ is less than Δ, where (F, ℓ) has the Δ-separation property. See Figure 4(d).

To round up the above three cases, it is convenient to call the special NULL value an "intersection rep" as well. Thus, an intersection algorithm for the input pair (F, ℓ) can return this NULL value in case $F \cap \ell = \emptyset$. The following lemma justifies our definition of reps:

LEMMA 11 *Suppose (F, ℓ) is an intersection rep. There F and ℓ has a unique intersection point p. Depending on the type of intersection rep, we deduce additional information:*

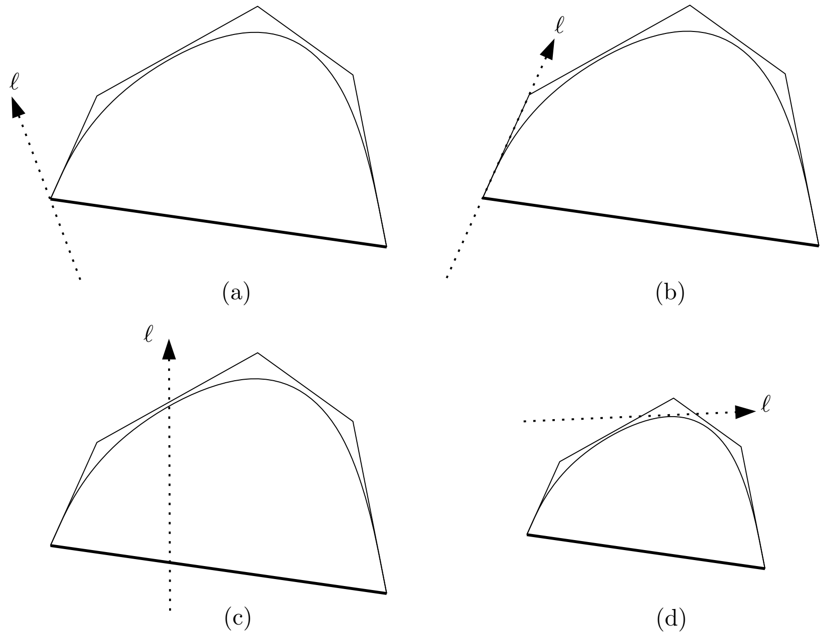


Figure 4: Intersection Representation for Line ℓ and Elementary curve F.

- *(Exact Type) p is an endpoint of F. Furthermore, p represents a tangential intersection iff ℓ contains the edge of the control polygon incident to p (see Figure 4(b)).*
- *(Transversal Type) p is a transversal intersection.*
- *(Tangential Type) p is a tangential intersection.*

In view of this lemma, if (F, ℓ) is an intersection rep and p is the unique point in $F \cap \ell$, we can write

$$p \sim \cap[F, \ell] \tag{11}$$

to indicate this relationship. Next consider the **Elementary Curve-Line Intersection Problem**: *Given ℓ and F, to compute a list of intersection reps, each rep corresponding to an intersection point of ℓ and F.* The output list has between 0 and 2 reps. We may assume F is nonlinear.

ALGORITHM A:
Input: Elementary curve F and line ℓ.
Output: List of intersection reps for the intersection of F and ℓ.

1 If ℓ does not intersect the $CH(F)$, there can be no intersection. Return(NULL).

2 If ℓ intersects an end point p of the base segment of F, but not the interior of $CH(F)$. We have the exact intersection at p. There are two subcases: (a) If ℓ contains the base segment, there are two exact intersection points. We split F into F_1 and F_2 and return$((F_1, \ell), (F_2, \ell))$. (b) Otherwise, we simply return$((F, \ell))$.

3 If ℓ intersects the relative interior of the base segment. This is a transversal intersection: return$((F, \ell))$.

4 If ℓ intersects the interior of $CH(F)$ and the diameter of P is less than Δ, we have a tangential intersection: return$((F, \ell))$.

5 Subdivide F into $F_1 = F[0, 1/2]$ and $F_2 = F[1/2, 1]$. We next have the several cases.

5.1 Suppose ℓ contains the base segment of F_i ($i = 1$ or 2). Note that i is unique. We recursively call (ℓ, F_i).

5.2 Else, if ℓ contains the point $F(1/2)$ we consider two subcases: (a) If ℓ does not intersect the relative interiors of $CH(F_1) \cup CH(F_2)$, we can return$((F_1, \ell))$. (b) If ℓ intesects the relative interior of $CH(F_i)$ ($i = 1$ or 2; again i is unique), then recursively call (F_i, ℓ).

5.3 Else, if ℓ intersects the base segments of both F_1 and F_2. We return$((F_1, \ell), (F_2, \ell))$.

5.4 Else, suppose ℓ intersects one of the convex hulls $CH(F_i)$ ($i = 1, 2$). This i is unique. We recursively call the algorithm on (F_i, ℓ).

5.5 At this point, we conclude that ℓ does not intersect F_1 or F_2. Return(NULL).

The algorithm relies on the following property:

LEMMA 12 *Assume ℓ does not intersect the base segment of F, and suppose F is subdivided into $F_1 = F[0, t]$ and $F_2 = F[t, 1]$ for some $0 < t < 1$. Then one of the following cases hold:*
(i) ℓ intersects the base segments of F_1 and F_2.
(ii) Else ℓ intersect the $CH(F_1)$ or $CH(F_2)$, but not both.
(iii) Else ℓ does not intersect either $CH(F_1)$ or $CH(F_2)$.

Proof. Assume (i)-(iii) does not hold. Only two other logical possibilities are not covered:
(iv) ℓ intersects only one of the two base segments of F_1, F_2.
(v) ℓ intersects both $CH(F_1)$ and $CH(F_2)$ without intersecting their base segments.

We can exclude (iv) because, when ℓ does not intersect the base segment of F, then it can only intersect F tangentially or in two distinct points. If intersecting tangentially, then ℓ cannot intersect any base segments; if intersecting in two distinct points, then it cannot intersect only one base segment.

We can exclude (v) because of the following observation. Form the tangent line T to F at $F(t)$. Then ℓ must intersect T to one side of $F(t)$. This means that ℓ cannot intersect the convex hull $CH(F_i)$ to the "other side" of ℓ. **Q.E.D.**

LEMMA 13 (CORRECTNESS) *Algorithm A terminates and detects all intersection reps.*

Proof. First we show partial correctness, i.e., assuming the algorithm terminates, the output is correct. Clearly, the outputs in Steps 1-3 are correct. The output of Step 4 is correct because of our intersection criterion. The correctness of Step 5 follows from Lemma 12. Also note that the algorithm returns between 0 and 2 intersection reps, with no redundancy.

It remains to show termination: assume the algorithm does not halt. Then our recursive algorithm have been invoked with the arguments (F_i, ℓ) for an infinite sequence $\{F_i : i \geq 0\}$ of curves where each F_{i+1} is a subcurve of F_i, and each convex hull $CH(F_i)$ intersect ℓ. By compactness, there is a point p such that $p \in F_i \cap \ell$ for all i. If p is a transversal intersection, then we will detect this in Steps 2 or 3 when F_i is sufficiently small. If p is a tangential intersection, then we would have detected this in Step 4 when F_i is sufficiently small. Thus we obtain a contradiction.

Q.E.D.

The line-curve intersection algorithm has two useful applications in the rest of our algorithms. We assume p and F in the following are explicitly represented in bigfloats.

- **Intersection Rep Refinement Problem**: Given an intersection rep (F, ℓ) where $F = F[a, b]$, we want to refine the rep into another intersection rep (F', ℓ) where $F' = F[a, (a + b)/2]$ or $F' = F[(a + b)/2, b]$.

  Let F be subdivided into F_0, F_1 at $F((a + b)/2)$. We need to decide which F_i to keep. For exact rep, it is trivial to determine the correct F_i. For transversal rep, it is the subcurve whose base segment intersects ℓ. For tangential rep, we may be lucky to find an exact rep, which is easily checked. Otherwise ℓ will intersect exactly one of the control polygons $P(F_i)$. This F_i is the desired F'.

- **Beneath or Beyond Test**: For points $q \neq v$, let $R(q, v)$ denote the ray originating at q and passing through v.

  The **Beneath-Beyond Test**, on input (F, q, v), will produce one of three outputs: ON if $p = q$, BEYOND if p lies on $R(q, v)$, and BENEATH otherwise.

This can be reduced to the line intersection, and intersection rep refinement computation above: first compute the intersection of F with the line through q, v. If there are no intersections, we return BENEATH. Otherwise, if q lies outside $CH(F)$, we can easily decide whether it is BENEATH or BEYOND. So suppose q lies in $CH(F)$. We then apply the previous refinement procedure for the intersection rep (F, ℓ) until either q lies outside $CH(F)$ or $\mathrm{diam}(CH(F))$ is smaller than the separation bound Δ_3 (see Theorem 3). In the latter case, we output ON; otherwise, it is straightforward to output BENEATH or BEYOND, appropriately.

6 Determining the Sign of the Alpha Angle

Our criterion (Theorem 10) for non-crossing intersection of an elementary pair (F, G) requires computing the sign of the angles $\alpha(0)$ and $\alpha(1)$. We now develop an adaptive procedure for this computation.

Let F be an elementary curve with the Bezier-type parametric form $F(t) = (F_1(t), F_2(t))$, and ℓ be a line with parametric form $L(u) = (cu + d, eu + f)$ for constants c, d, e, f. The slope angle of ℓ is defined to be θ_ℓ where

$$\cos(\theta_\ell) = \frac{c}{\sqrt{1+c^2}}, \qquad \sin(\theta_\ell) = \frac{e}{\sqrt{1+c^2}}. \tag{12}$$

We assume $e > 0$. Thus ℓ is non-horizontal. We also see that $\theta_\ell \in (0, \pi)$. The normal angle $\theta_F(t)$ at $F(t)$ is given by

$$\cos(\theta_F(t)) = \frac{-F_2'(t)}{\sqrt{F_1'^2(t) + F_2'^2(t)}}, \qquad \sin(\theta_F(t)) = \frac{F_1'(t)}{\sqrt{F_1'^2(t) + F_2'^2(t)}} \tag{13}$$

where $F_i'(t)$ is the derivative with respect to t.

Let ℓ intersect F at the point $F(t^*)$ for some t^*. We want to determine the sign of the angle

$$\alpha(t^*) = \theta_F(t^*) - \theta_\ell.$$

Assume $F_1'(t) \geq 0$. It follows that $\theta_F(t) \in (0, \pi)$ and $\alpha(t) \in (-\pi, \pi)$. Thus $\alpha(t^*)$ has the same sign as the expression

$$\sin(\alpha(t^*)) = \sin(\theta_F(t^*))\cos(\theta_\ell) - \cos(\theta_F(t^*))\sin(\theta_\ell) \tag{14}$$

Plugging (12) and (13) into (14) and clearing denominators, this reduces to the sign of the quantity

$$s_0 := cF_1'(t^*) + eF_2'(t^*). \tag{15}$$

On the other hand, let $F(t^*) = L(u^*)$ where u^* satisfies $u^* = (F_1(t^*) - d)/c = (F_2(t^*) - f)/e$, or

$$eF_1(t^*) - de = cF_2(t^*) - cf. \tag{16}$$

Thus s_0 is a zero of the resultant

$$\mathtt{res}_t(\mathcal{B}_1(s,t), \mathcal{B}_2(s,t)) \tag{17}$$

where

$$\mathcal{B}_1(s,t) = s - cF_1'(t) - eF_2'(t), \qquad \mathcal{B}_2(s,t) = eF_1(t) - cF_2(t) + (cf - de). \quad (18)$$

Actually, $\mathcal{B}_2(s,t)$ does not depend on s. The degrees of $\mathcal{B}_1$ and $\mathcal{B}_2$ are $m' = \max\{1, m-1\}$ and m, respectively.

THEOREM 14 *Suppose the control polygon of F are represented by points whose coordinates are L-bit floats, and suppose c, d, e, f are L-bit floats. If $s_0 \neq 0$, then*

$$|s_0| \geq (6m128^L 9^m)^{-m}.$$

Proof. We use the generalized Hadamard bound, arguing as in the proof of Lemma 5. Let ν_i be the $(2,1)$-norm of $\mathcal{B}_i(s,t)$ with respect to t, for $i = 1, 2$. We cannot directly use $\mathcal{B}_i$ as its coefficients may not be integer. Instead, use $2^L\mathcal{B}_1$ and $4^L\mathcal{B}_2$, which have integer coefficients. The corresponding norms will be $2^L\nu_1$ and $4^L\nu_2$. Thus the 2-norm of the resultant (17) is bounded by $(2^L\nu_1)^m(4^L\nu_2)^{m'}$. Since s_0 is the root of this resultant, if $s_0 \neq 0$ then

$$|s_0| \geq \frac{1}{(2^L\nu_1)^m(4^L\nu_2)^m}. \quad (19)$$

It remains to bound the ν_i's. By Lemma 4, we see that $F_1(t) = \sum_{k=0}^m a_k t^k$ where $|a_k| \leq 2^{L+k}\binom{m}{k}$. Calculating as in Lemma 5, we conclude that $\|F_1(t)\|_1 \leq 2^L 3^m$. Assuming $L \geq 1$, the same trick as in the proof of Lemma 5(ii) implies

$$\|F_1(t)\|_2 \leq 2^L 3^m - 1.$$

So $\|F_1'(t)\|_2 \leq m(2^L 3^m - 1)$ and $\|eF_1'(t)\|_2 \leq m2^L(2^L 3^m - 1)$. Similarly, $\|cF_2'(t)\|_2 \leq m2^L(2^L 3^m - 1)$. It follows that

$$\nu_1 \leq 1 + 2m2^L(2^L 3^m - 1) < 2m4^L 3^m.$$

Similarly,

$$\nu_2 \leq 2 \cdot 2^L(2^L 3^m - 1) + 2 \cdot 4^L \leq 3^{m+1}4^L.$$

The result follows by plugging into (19). **Q.E.D.**

Application 1: Alpha Sign Problem. The problem is this: given an explicit line ℓ and $G_0 \sim Bezier[G^*, s_0, t_0]$, compute the sign of the angle between ℓ and the normal through G^* at $p \sim \cap[G_0, \ell_0]$. The point p is equal to $G^*(t^*)$ for some $t^* \in [s_0, t_0]$. So the problem amounts to computing the sign of the function

$$g(t) = cG_1'(t) + eG_2'(t)$$

at $t = t^*$, where $G^*(t) = (G_1(t), G_2(t))$ and $G_i'(t)$ is the derivative of $G_i(t)$ relative to t, the slope of the line ℓ_0 is c (cf. (15)). The main issue in determining the sign of $g(t^*)$ is that t^* is not a direct number: we want to determine this sign adaptively, that is, by using better and better approximations to t^*. This is given by Algorithm B.

ALGORITHM B:

Input: an elementary curve $G_0 \sim Bezier[G^*, s_0, t_0]$ and line ℓ. G_0 and ℓ intersect at some (unknown) point $G_0(t^*)$. The control polygon for G^* and ℓ are defined by L-bit floats.

Output: Sign of the angle $\alpha(t^*) = \theta_{G_0}(t^*) - \theta_\ell$.

1 Evaluate all the partial derivatives $g^{(i)}(t)$ of $g(t)$ at $t = s_0$.

2 Bound the error $|g(t^*) - g(s_0)|$ using Taylor's expansion, by computing an upper bound to the quantity

$$\varepsilon = \sum_{i \geq 1} \left| g^{(i)}(s_0) \right| \frac{\delta^i}{i!}.$$

where $\delta = t_0 - s_0$.

3 If $|g(s_0)| \geq \varepsilon$, return($\texttt{sign}(g(s_0))$).

4 If $-\lg \varepsilon \geq m(\lg 6 + \lg m + 7L + m \lg 9)$, return(0).

5 Otherwise, refine the intersection representation to $p \sim \cap[G_1, \ell]$ where $G_1 \sim Bezier[G^*, s_1, t_1]$ and either s_1 or t_1 is equal to $(s_0 + t_0)/2$. Rename $(s_0, t_0) \leftarrow (s_1, t_1)$ and go back to step 1.

REMARKS:
(i) All the indicated operations can ultimately be reduced to computations with direct numbers, so that the algorithm will yield the correct decision. For instance, the computed version of ε will be an upper bound on the expression we write in Step 2. Similarly, the comparison in Step 4 will actually be reduced to some comparison that implies the inequality $-\lg \varepsilon \geq m(\lg 6 + \lg m + 7L + m \lg 9)$ which is explicitly mentioned in Step 4. We write the exact mathematical expressions because this is easier for understanding the underlying mathematical goals.
(ii) In implementation, the recursive steps in Algorithm B can benefit from remembering intermediate computed values.
(iii) Step 1 in the algorithm can be sped up using techniques from the field of automatic differentiation so that the cost of evaluating all partial derivatives is not much more than the evaluation of $g(s_0)$ itself.
(iv) The complexity of the algorithm is adaptive (only in the unlikely case of a return in Step 4 is the worst case behavior achieved).

Application 2: Determining tangency at a known point. The following problem arises in the next section. Since its solution reduces to Algorithm B, we present the material now.

The problem is this: given $G \sim Bezier[G^*, s_0, t_0]$ and two points p, q. It is known that $p \in G$, and we want to determine whether the line $\ell(p, q)$ through p and q is a tangent line.

We may assume $\ell(p, q)$ is not vertical. Let $L(u) = (cu + d, eu + f)$ be the parametric line through p which is perpendicular to $\ell(p, q)$. Clearly,

$$d = p.x, \quad f = p.y$$

and c, e must satisfy the equation

$$c \cdot (p.x - q.y) + e \cdot (p.y - q.y) = 0.$$

We can choose $\delta = \pm 1$ so that $e = \delta \cdot (p.x - q.x) > 0$ and $c = -\delta \cdot (p.y - q.y)$. Thus $\delta = 1$ if $p.x > q.x$ and $\delta = -1$ otherwise.

If α is the angle between $L(u)$ and the normal to G at p, then $\alpha = 0$ iff $\ell(p, q)$ is a tangent. Thus, our problem amounts to deciding if $\mathtt{sign}(\alpha) = 0$.

We can use Algorithm B for this decision, but we need to determine the applicable root bound: let us assume that all the input numbers are L-bit floats. Then d, f are L-bit floats, but we see that c, e are $2L$-bit floats. Thus in Step 4 of Algorithm B, we simply replace L with $2L$.

7 Coupling Process – First version

In this section, assume that F, G are elementary Bezier curves satisfying the Δ-separation property, and such that $F \cup G$ has diameter less than Δ. This implies that $|F \cup G| \leq 1$. Our goal is to determine whether they intersect by ultimately reducing them to NIC, at least implicitly. This process is called the **coupling process**.

OVERVIEW. By way of motivation, note that if we try to directly apply the NIC, we must carefully subdivide F, G until some pair (F', G') of subcurves that forms a couple. This seems difficult to achieve in general. But if the pair (F, G) is a "micro pair" which guarantees that they have at most one intersection, then things are greatly simplified. But still, a direct couple may be hard to find. So we give an "extended NIC" criterion that is derived from the NIC.

The first step is to detect if F and G have crossing intersection. If so, we are done. So assume otherwise. Up to symmetry, we could reduce the situation to the case where F is A-elementary and G is **above** F. *By definition, G is above F means some open interval of G is contained in $A(F)$, the upper swept region of F.* Next, we need to use binary search to find a t in the domain of $F = F[a, b]$ such that $a_F(t)$ intersects G. This search will succeed because we know that G is above F. A slight complication might arise because $a_F(t)$ might intersect G in two points. In any case, we can reduce the search to one or two **half-couples**, i.e., one end normal, but not the other end normal, of F passes through an endpoint of G. Thus half-couples satisfy only "half" the conditions for coupled pairs. Finally, we adapt our NIC to half-couples. This completes our coupling process.

Here are more details.

1. Let $S(F)$ denote the **vertical slab** containing F such that the boundary of $S(F)$ are the two vertical lines through the ends of F. We view $S(F)$ as a closed set. Clearly F and G intersect within the strip $S(F) \cap S(G)$, if they intersect at all. So we would like to check if $S(F) \cap S(G)$ is empty. We can reduce this check to the Line Intersection Test of the previous section. More precisely, we first check if the vertical line though each end point of

F intersects G, and also if the vertical line through each end point of G intersects F. Thus there are 4 tests, which we perform sequentially (in any order). We stop as soon as we find two vertical lines with intersection. Otherwise, we conclude that $S(F) \cap S(G)$ is empty.

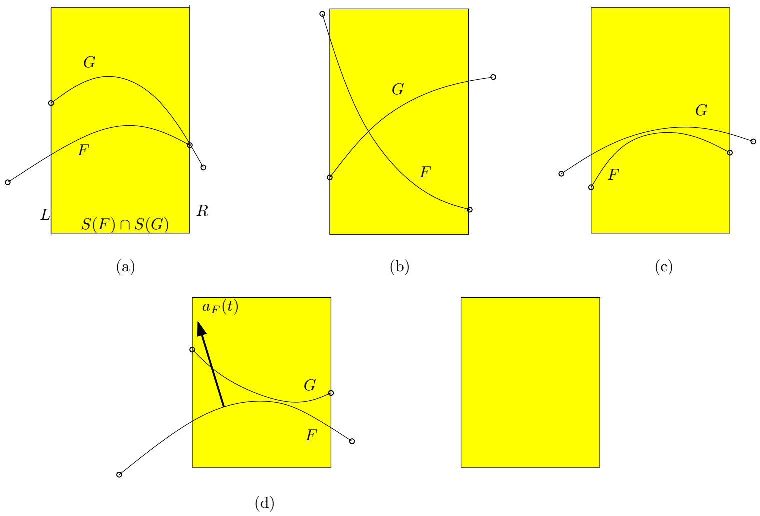


Figure 5: (a) Endpoint Intersection (b) F and G are crossing (c) F below G, with F is A-elementary.

2. Assuming $S(F) \cap S(G)$ is non-empty, let the vertical lines L and R (resp.) be the left and right boundaries of $S(F) \cap S(G)$. By applying the Beneath-Beyond tests to the endpoints of F or G, we can determine whether F is above, below or equal to G along the lines L, R. In particular, we would know whether F and G intersects along L or R. If so, we are done. See Figure 5(a). So assume otherwise.

3. We say that F and G are **crossing** if F lies above G along L iff F lies below G along R. Clearly, the F and G are crossing, then we know that F and G has transversal intersection, and we are done. See Figure 5(b). Otherwise, without loss of generality, assume F is below G along L as well as along R. This implies that G **lies above** G as defined before. The following three cases are exhaustive:
(i) F is A-elementary
(ii) G is B-elementary
(iii) F is B–elementary and G is A-elementary.

4. Clearly, in case (iii), there can be no intersection. We only need to consider case (i), as case (ii) is symmetrical. This is illustrated in Figure 5(c,d). Assuming $F = F[a, b]$ and $G = [c, d]$ in Bezier parametrizations, our first

goal is to find a $t \in [a, b]$ such that the the upper half-normal $a_F(t)$ intersects G. We can do this using a binary search procedure: in general, let $[a', b'] \subseteq [a, b]$ is the current interval of our search for t. Initially $[a', b'] = [a, b]$. Setting $c' = (a' + b')/2$, we check whether the half-normal $a_F(c')$ intersects G. If so, we output $t = c'$. Otherwise, it is easy to determine whether we should update $[a', b']$ to $[a', c']$ or $[c', b']$. This procedure is sure to halt.

5. Suppose we have found $t \in [a, b]$ such that $a_F(t)$ intersects G. In general, a half normal $a_F(t)$ might intersect G at two points.

   (a) First assume G is A-elementary. Then we see that $a_F(t)$ must intersect G at a unique point $G(s)$ for some s. We can therefore reduce the search for intersection in each of the following pairs: $(F[a, t], G[c, s])$ and $(F[a, t], G[s, d])$. Note that $(F[a, t], G[c, s])$ is a half-couple because $a_F(t)$ passes though the endpoint $G(s)$ of $G[c, s]$. Similarly, $(F[a, t], G[s, d])$ is a half-couple.

   (b1) Next, assume G is B-elementary. If $a_F(t)$ intersects G at only one point $G(s)$, we can discard either $G[c, s]$ or $G[s, d]$, and reduce the search to the pair $(F[a, t], G[c, s])$ or to $(F[t, b], G[s, d])$. Both pairs are half-couples.

   (b2) But $a_F(t)$ can also intersect G at two points, $G(s)$ and $G(s')$ for some $c \leq s \leq s' \leq d$. By symmetry, assume that $a_F(t)$ points to the left of the vertical direction. In this case, we can reduce the search to either the half-couple $(F[a, t], G[s, s'])$ or the half-couple $(F[t, b], G[s', d])$. The correct case is determined by the angle at which $a_F(t)$ intersects $G(s')$.

   Remark that we do not represent $G[c, s(t)]$ or $G[s(t), d]$ explicitly because $s(t)$ is not generally rational. Instead, we just remember that $G(s(t))$ as the intersection of $n_F(t)$ and G where t is a bigfloat number.

6. It remains to develop "intersection criteria" for half-couples. This turns lemma shows how we can apply NIC even for half-couples:

   Lemma 15 *Suppose $(F[a, b], G[c, d])$ is a half-couple in that $a_F(a)$ passes through $G(c)$ and $a_F(b)$ does not intersect G. Moreover, the end point $G(d)$ lies above F. Let $b_G(d)$ denote the lower half-normal at $G(d)$. We define $\theta_G(d)$ to be the angle that the upward normal at $G(d)$ makes with the positive x-axis (as before). We have five possibilities:*
   (i) $b_G(d)$ intersects $a_F(a)$.
   (ii) $b_G(d)$ intersects $a_F(b)$.
   (iii) $b_G(d)$ intersects F at $F(t)$. Moreover $\theta_F(t) - \theta_G(d)$ is positive.
   (iv) $b_G(d)$ intersects F at $F(t)$. Moreover $\theta_F(t) - \theta_G(d)$ is negative.
   (v) $b_G(d)$ intersects F at $F(t)$. Moreover $\theta_F(t) = \theta_G(d)$.
   Let t_0 be chosen such that $a_F(t_0)$ intersects $G(d)$. Then in cases (i) and (iii), $\alpha_{F,G}(t_0) > 0$. In cases (ii) and (iv), $\alpha_{F,G}(t_0) < 0$. In case (v), $\alpha_{F,G}(t_0) = 0$.

Proof. easy... **Q.E.D.**

7. Using the above lemma, we can now apply the NIC to half-couples as follows: we compute the intersection of $b_G(d)$ with F, $a_F(a)$ and $a_F(b)$. This will determine the angle $\alpha_{F,G}(t_0)$, by the previous lemma. But we also know $\alpha_{F,G}(a)$. Thus, by the NIC, there is intersection iff

$$\alpha_{F,G}(a)\alpha_{F,G}(t_0) \leq 0.$$

8 Critical Points and Non-Elementary Curves

We now address the issue of non-elementary Bezier curves. A Bezier curve $F[0,1]$ can be subdivided into finitely many elementary subcurves. The points $F(t)$ at which such subdivisions occur are the "critical points".

More precisely, let $F_x(t), F_y(t)$ be the coordinate functions of F (i.e., $F(t) = (F_x(t), F_y(t))$, and let and F', F'_x, F'_y denote derivatives with respect to t. A point $F(t)$, $0 < t < 1$, on $F = F[0,1]$ is **critical** if one of the following conditions hold (cf. Kim and Lee [10]):

- (i) $F(t)$ is **stationary**, i.e., $F'_x(t) = F'_y(t) = 0$;
- (ii) $F(t)$ is **x-extreme**, i.e., $F'_x(t) = 0$, $F'_x(t^-)F'_x(t^+) < 0$ and $F'_y(t) \neq 0$;
- (iii) $F(t)$ is an **inflection point**, i.e., $H_F(t) = 0$ and $H_F(t^+)H_F(t^-) < 0$, where
$$H_F(t) := F'_x(t)F''_y(t) - F'_y(t)F''_x(t).$$

Lemma 16 *If F has no critical points in its relative interior then F is elementary. Conversely, if F is elementary, then it has no x-extreme or inflection points in its relative interior.*

The proof may be omitted. This lemma suggests that an elementary curve may have stationary points. Indeed, if $P(F) = ((-1,0),(0,0),(0,0),(1,0))$ then F is elementary (in fact, a straight line segment) but $F(1/2)$ is a stationary point.

Unfortunately, the parameter t at which $F(t)$ is critical is most likely an irrational number. Hence we do not propose to subdivide Bezier curves into elementary parts by subdividing at critical points. We will insist that we only subdivide at t which are bigfloat values, as before. Moreover, we will subdivide the Bezier curve into subcurves that are either elementary or contain at most one critical point. We may call a curve that contains exactly one critical point an $K-$, $C-$ or $S-$**curve** depending on whether the critical point is a stationary, x-extreme or inflexion point. Now we could study the interactions between these new kinds of curves among themselves and with elementary curves. But this seems like a very tedious approach. Instead, we will appeal to new separation bounds, based on a number $\Delta^* > 0$ which we will explain.

Lemma 17 *Let $P(F) = (p_0, \ldots, p_m)$ where coordinates of the p_i's are (L, ℓ)-bit floats.*
(i) If $F(t) = (F_x(t), F_y(t))$ is a stationary or x-extreme point, then $F_x(t)$ and $F_y(t)$ has degree $\le m-1$ and 2-norm at most $(4^L 9^m m)^m$.
(ii) If $F(t)$ is an inflexion point, then $F(t)$ has degree $\le 2m-3$ and 2-norm at most $(2m^4 16^L 81^m)^m$.

Proof. (i) If $F(t) = (X, Y)$ is stationary or x-extreme, then X satisfies the equation $R(X) = \mathtt{res}_t(F_x(t) - X, F'_x(t))$. If $F_x(t) = \sum_{i=0}^m a_i t^i$, then $R(X)$ is the determinant of the following matrix M',

$$M' = \begin{bmatrix} a_m & a_{m-1} & \cdots & a_1 & a_0 - X & \cdots & \\ & a_m & \cdots & a_2 & a_1 & \cdots & \\ & & \ddots & & & \cdots & \\ & & & a_m & a_{m-1} & \cdots & a_0 - X \\ ma_m & (m-1)a_{m-1} & \cdots & 2a_2 & a_1 & & \\ & ma_m & \cdots & 3a_3 & 2a_2 & \cdots & \\ & & \ddots & & & \cdots & \\ & & & ma_m & (m-1)a_{m-1} & \cdots & a_1 \end{bmatrix} \tag{20}$$

Similar to the proof of Theorem 6, we consider the matrix

$$M_2 = \begin{bmatrix} 2^{L+m}\binom{m}{0} & 2^{L+m-1}\binom{m}{1} & \cdots & 2^{L+1}m & 2^L+1 & & \\ & 2^{L+m}\binom{m}{0} & \cdots & 2^{L+2}\binom{m}{2} & 2^{L+1}m & & \\ & & \ddots & & & \ddots & \\ & & & 2^{L+m}\binom{m}{0} & 2^{L+m-1}\binom{m}{1} & \cdots & 2^L+1 \\ m2^{L+m}\binom{m}{0} & (m-1)2^{L+m-1}\binom{m}{1} & \cdots & 2.2^{L+2}\binom{m}{2} & 2^{L+1}m & & \\ & m2^{L+m}\binom{m}{0} & \cdots & 3.2^{L+3}\binom{m}{3} & 2.2^{L+2}\binom{m}{2} & & \\ & & \ddots & & & \ddots & \\ & & & m2^{L+m}\binom{m}{0} & (m-1)2^{L+m-1}\binom{m}{1} & \cdots & 2^{L+1}m \end{bmatrix}. \tag{21}$$

The first $m-1$ rows of M_2 have 2-norms $2^L 3^m$ (as in Lemma 5). The last m rows of M_2 have 2-norms $\le m2^L 3^m$. Applying the generalized Hadamard bound, $\|R(X)\| \le (2^L 3^m)^{m-1}(m2^L 3^m)^m$, which is $2^{(2m-1)L}3^{2m^2-m}m^m < (4^L 9^m m)^m$, as we claim.
(ii) We proceed as in (i), but the matrix M' now has $2m-3$ rows of the coefficients of $F_x(t) - X$, and has m rows of the coefficients of $F'_x(t)F''_y(t) - F''_x(t)F'_y(t)$. Thus the degree of $R(X) = \det(M')$ is $2m-3$. We have $\|F'_x(t)\| \le m2^L 3^m$, $\|F''_y(t)\| \le m^2 2^L 3^m$. Hence $\|F'_x(t)F''_y(t)\| \le m(m2^L 3^m)(m^2 2^L 3^m) = m^4 4^L 9^m$ using the fact that $\|AB\| \le m\|A\| \cdot \|B\|$ where A, B have degrees $\le m$. Hence $\|F'_x(t)F''_y(t) - F''_x(t)F'_y(t) \le 2m^4 4^L 9^m$. Thus $\|R(X)\| \le (2^L 3^m)^{2m-3}(2m^4 4^L 9^m)^m < (2m^4 16^L 81^m)^m$. **Q.E.D.**

Let α, β be algebraic numbers, of degrees at most m, n and 2-norms at most a, b, respectively. If $m \le n$, then

$$|\alpha - \beta| > (2^{mn+m} a^n b^m)^{-1} \tag{22}$$

([18, Lemma 35, Chap. 6]). This gives us:

Corollary 18 *Suppose p, q are two distinct critical points of F. If $2 \le m \le n$ then*

$$|p.x - q.x| \ge \Delta_5 := (16^{m+2} 256^L 81^{2m} m^5)^{-m}.$$

A similar inequality holds for $|p.y - q.y|$.

Proof. Directly apply (22) on the size bounds on $p.x$ and $q.x$, from the previous lemma. **Q.E.D.**

Let us choose Δ^* to be the minimum of the bounds Δ_1, Δ_2, Δ_4 and Δ_5. Call F a **critical curve** if F contains a critical point and the diameter of F is less that Δ^*). We will need procedures to check if a curve F is a critical curve. First we generalize Theorem 3 to the case where q is any algebraic point:

Theorem 19 *Let $q = (q.x, q.y)$ where $q.x, q.y$ are algebraic numbers with 2-norm $\le c$ and degree $\le d$. If $A = 0$ does not contain a circle centered at q then the distance from q to A is at least $\Delta_4(m, a, L, c, d) = (2^{3/2} NK)^{-D} 2^{-12m^2d^2}$ where*

$$K = \max\{\sqrt{13}, 4ma, c\}, \quad N = \binom{5 + 2m + 2d}{5}, \quad D = m^2d^2(3 + \frac{4}{m} + \frac{4}{d}).$$

Proof. We may assume p is a closest point on A to q and $h = \|p - q\|$. Thus q, p, h satisfy the system,

$$\begin{aligned}
\mathcal{A}_1 &: A(p) = 0,\\
\mathcal{A}_3 &: h^2 - \|p - q\|^2 = 0,\\
\mathcal{A}_4 &: \left\langle (\frac{dA}{dy}(p), -\frac{dA}{dx}(p)), (p - q) \right\rangle = 0,\\
\mathcal{A}_9 &: P(q.x) = 0,\\
\mathcal{A}_{10} &: Q(q.y) = 0,
\end{aligned}$$

where $P(X), Q(X)$ are polynomials of degree $\le d$ and 2-norm $\le c$ satisfied by $q.x$ and $q.y$, respectively. This system is zero-dimensional since $A = 0$ does contain a circle centered at q. We have $\|\mathcal{A}_1\|_2 = a$, $\|\mathcal{A}_3\|_2 \le \sqrt{13}$, $\|\mathcal{A}_4\|_2 \le 4ma$, $\|\mathcal{A}_9\|_2 \le c$, and $\|\mathcal{A}_{10}\|_2 \le c$. The result follows by plugging into Fact 1. **Q.E.D.**

Detection of Stationary Points. Let $P(F) = (p_0, \ldots, p_m)$ be the control polygon of F. Then

$$F(t) = \sum_{i=0}^{m} p_i B_m^i(t)$$

where $B^i_m(t) = \binom{m}{i} t^i (1-t)^{m-i}$. It is well-known that the derivative with respect to t is given by

$$\begin{aligned} F'(t) &= m \sum_{i=1}^{m} [p_i - p_{i-1}] B^{i-1}_{m-1}(t) \\ &= m \sum_{i=1}^{m} \nabla p_i B^{i-1}_{m-1}(t) \end{aligned}$$

where we define $\nabla p_i := p_{i+1} - p_i$. Also, let $\nabla P(F) = (\nabla p_1, \ldots, \nabla p_m)$. Thus, we see that $m\nabla P(F)$ is the control polygon for the curve $F'(t)$, known as the **hodograph** of F. Thus F contains a stationary point iff $F'(t)$ passes through the origin. Figure 6 shows a cubic Bezier whose control polygon is $((0,0),(1,0),(0.5,-0.5),(0.5,0.5))$.

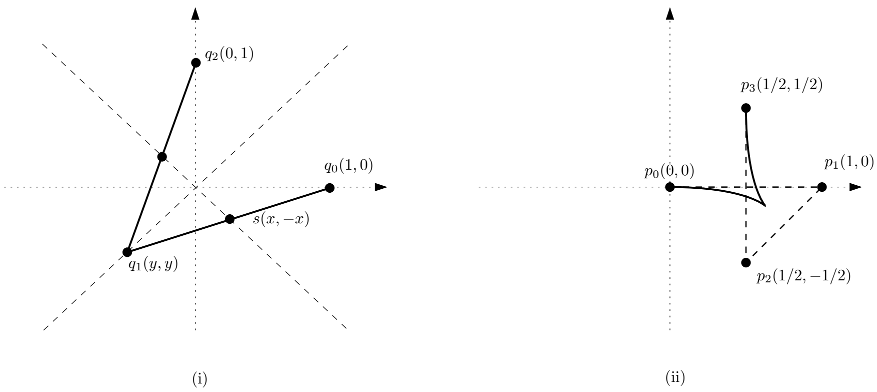


Figure 6: Singular cubic Bezier curve: (i) hodograph, (ii) control polygon

THEOREM 20 *Let* $\mathrm{diam}(F) < \Delta_3(m-1, ma, 2)/m$. *Then F contains a stationary point iff the convex hull of $\nabla P(F)$ contains the origin* $(0,0)$.

Proof. If F contains a stationary point, then clearly its hodograph passes through $(0,0)$. Hence the convex hull of $m\nabla P(F)$ contains $(0,0)$. Equivalently, the convex hull of $\nabla P(F)$ contains $(0,0)$. Conversely, suppose F does not contain a stationary point. Note that F' has degree $m-1$ and $\|F'\|_2 \le m\|F\|_2 = ma$. By Theorem 3, the distance from $(0,0)$ to the curve F' is at least $\Delta_3(m-1, ma, 2)$ since $(0,0)$ is a 2-bit float. Thus the distance from $(0,0)$ to F'/m is at least $\Delta_3(m-1, ma, 2)/m$. On the other hand, the diameter of $\nabla P(F)$ is at most the diameter of F, i.e., less than $\Delta_3(m-1, ma, 2)/m$. This means $(0,0)$ could not be inside the convex hull of $\nabla P(F)$. **Q.E.D.**

Detection of Extreme Points. If $F(t)$ is an x-extreme point of F, then $F'(t) = 0$.

THEOREM 21 *Suppose* $\mathrm{diam}(F) < \Delta^*$. *Then* F *contains an* x*-extreme point iff* $(\nabla p_1).x(\nabla p_m).x \leq 0$ *and* $(\nabla p_1).y(\nabla p_m).y > 0$.

Proof. If $(\nabla p_1).x(\nabla p_m).x \leq 0$ and $(\nabla p_1).y(\nabla p_m).y > 0$ then clearly F contains an x-extreme point. Conversely, suppose $F(t)$ is an x-extreme point, then we know that both $F[0,t]$ and $F[t,1]$ lies strictly to one side of the vertical line through $F_x(t)$. Since $\mathrm{diam}(F)$ is small, there are no other critical points, not even y-extreme points. Thus, the two subcurves are elementary. Now it is clear that the first and last edges ∇p_1 and ∇p_m on the control polygon $\nabla P(F)$ has the claimed properties. **Q.E.D.**

Detection of Inflection Points. If p, q, r are planar points, let $\det(p, q, r)$ be the determinant of the 3×3 matrix whose rows are p^h, q^h, r^h, where $p^h = (p.x, p.x, 1)$. Also let $\det(p, q)$ denote $p.xq.y - p.yq.x$.

THEOREM 22 *Suppose* $\mathrm{diam}(F) < \Delta^*$. *Then* F *contains an inflexion point iff* $\det(p_0, p_1, p_2)\det(p_{m-2}, p_{m-1}, p_m) < 0$.

Proof. We make the connection between $\det(p_0, p_1, p_2)$ and inflection points. Let $H(t) = F'_x(t)F''_y(t) - F''_x(t)F'_y(t)$. Note that $F'(0) = p_1 - p_0$ and $F''(0) = p_2 - 2p_1 + p_0$. Hence $H(0) = \det(p_1 - p_0, p_2 - 2p_1 + p_0) = \det(p_1 - p_0, p_2 - p_1)$. $H(0) = -\det(p_0 - p_1, p_2 - p_1) = \det(p_1, p_0, p_2) = -\det(p_0, p_1, p_2)$. Similarly, $H(1) = -\det(p_{m-2}, p_{m-1}, p_m)$. Therefore, $\det(p_0, p_1, p_2)\det(p_{m-2}, p_{m-1}, p_m) < 0$, there must be some t such that $H(t) = 0$, and this would be an inflection point. Conversely, if $F(t)$ is an inflection point, then $H(t^-)H(t^+) < 0$. Since $\mathrm{diam}(F)$ is small, there are no other inflection points and therefore we conclude that $H(0)H(1) < 0$. **Q.E.D.**

To summarize, if F is a critical curve, and (F, G) is a candidate pair (i.e., the convex hulls of their control polygons intersect), we can conclude that the critical point $F(t)$ in F must belong to G. This is a unique intersection. So we can output this pair.

9 Intersection Algorithm

We now present the global algorithm for intersecting non-elementary Bezier curves F and G. We design the algorithm so that the generic intersection algorithm (see Introduction) is naturally embedded as the first phase of our algorithm. In this way, a comparison of our algorithm with this standard approach is easier. Simplicity rather than practical efficiency is the aim of the following description. In practice, some of our tests ought to be preceded by partial, but cheaper, criteria to filter out the easy cases.

We have two work queues Q_0 and Q_1. These queues may be just a list of candidate pairs (F_i, G_i) ($i \in 1, 2, 3, \ldots$). Recall that a candidate pair (F, G) is called a **micro pair** if the diameter of $CH(F) \cup CH(G)$ is less than Δ^*, and called a **macro pair** otherwise. We call Q_0 and Q_1 the **macro queue** and **micro queue**, so-called because they contain macro pairs and micro pairs, respectively. The algorithm also operates in two phases.

The macro phase: This is essentially the generic intersection algorithm, and operates on the macro queue Q_0. For each pair $(\widehat{F}, \widehat{G})$ extracted from Q_0, we subdivide the "larger" curve (say $\widehat{F}$) into two $\widehat{F}_0$ and $\widehat{F}_1$ at the point $\widehat{F}(1/2)$, and form pairs $(\widehat{F}_0, \widehat{G})$ and $(\widehat{F}_1, \widehat{G})$. The criterion for "larger" is flexible, and may be simply the length of the diagonal of the bounding box. A pair is discarded if it is not a candidate pair. Otherwise, we place it into Q_0 or Q_1 depending on whether it is a micro pair or not. If we did nothing else in the macro phase, we would still be correct. But in practice, we should perform other partial acceptance or rejection criteria. In particular, we should apply the following transversal intersection acceptance test: output a pair (F, G) if it satisfies the following conditions:

- The base segment $[a_F, b_F]$ of F and the base segment $[a_G, b_G]$ of G intersect.
- Both a_F, b_F do not lie in $CH(G)$, the convex hull of the control polygon of G; conversely, a_G, b_G do not lie in the $CH(F)$.
- The angular ranges for their normals are disjoint.

To show the correctness of this criteria, we first note that F and G must intersect: the first two criteria implies that $CH(F)$ and $CH(G)$ intersect in a convex region whose boundary can be divided into $2n$ parts, $(A_1, B_1, A_2, B_2, \ldots, A_n, B_n)$ listed in some circular order, $n \geq 2$, and $A_1 \cup A_2 \cup \cdots \cup A_n$ is part of the boundary of F, and $B_1 \cup B_2 \cup \cdots \cup B_n$ is part of the boundary of G. Now the base segment $[a_F, b_F]$ connects a point of some B_i with a point in B_j $(1 \leq i < j \leq n$, and similarly $[a_G, b_G]$ connects a point of some A_k with a point in A_ℓ $(1 \leq k < \ell \leq n)$. Since the base segments intersect, we have $k \leq i < \ell \leq j$ (i.e., A_k, B_i, A_ℓ, B_j interleave). It follows that F connects B_i to B_j and G connects A_k to A_ℓ. This implies the intersection of F and G. To show that there is at most one intersection, we use the last property: suppose $F \cap G$ contains two points $p \neq q$. We may assume (a_F, p, q, b_F) forms a convex quadrilateral inside $CH(F)$. Similarly we may assume (b_G, p, q, b_G) forms a convex quadrilateral inside $CH(G)$. But the norm

The micro phase: Now we operate on Q_1. For each pair $(\widehat{F}, \widehat{G})$ extracted from Q_1, we apply the **micro procedure** to it. This micro procedure will determine whether $\widehat{F}, \widehat{G}$ has a intersection, or they do not intersect. Note that if they intersect, the intersection point is unique.

1. STEP 0. We check if the endpoints of $\widehat{F}$ and $\widehat{G}$ are intersection points. This uses our separation bound between a point with floating point coordinates from a curve, in case the point is not not on curve. If any of these points are intersection points, we output the pair $(\widehat{F}, \widehat{G})$ with an indication about which endpoints are in the intersection.

2. STEP 1. We check if the relative interior of $\widehat{F}$ or $\widehat{G}$ contain critical points. (It does not matter if the endpoints are critical.) Let us focus on $\widehat{F}$. In the previous section, we showed how to determine if the critical point is in the interior of $\widehat{F}$. Assume we have now tested if $\widehat{F}$ and/or $\widehat{G}$ contain critical points. If both contain critical points, then $\widehat{F} \cap \widehat{G}$ is this critical point. We output $(\widehat{F}, \widehat{G})$ and halt. If neither contain critical points, we have found two elementary curves, and we can apply the coupling process from a previous section. Otherwise, exactly one of $\widehat{F}$ or $\widehat{G}$ has a critical point, and we proceed to STEP 2.

3. STEP 2. W.l.o.g., assume $\widehat{F}$ contains a critical point q in its relative interior, and $\widehat{G}$ is elementary. If $\widehat{F}$ intersects $\widehat{G}$, then q must be this intersection point. Note that we know whether q is stationary, x-extreme or an inflexion. We now keep subdividing $\widehat{F}$ into smaller and smaller subcurves, always maintaining q in the interior of the subcurves. More precisely, we test the point $\widehat{F}(1/2)$ lies on $\widehat{G}$. If so, we are done. Otherwise, we test the 2 subcurves $\widehat{F}[0, 1/2], \widehat{F}[1/2, 1]$ to see which of them contains a critical point in its relative interior. If $\widehat{F}[0, 1/2]$ has no critical point, we can use the coupling process on the pair of elementary curves $(\widehat{F}[0, 1/2], \widehat{G})$. If $\widehat{F}[0, 1/2]$ has a critical point, we continue the subdivision of this subcurve. We treat $\widehat{F}[1/2, 1]$ similarly.

   Recall that the coordinates of q has a certain degree bound d, and a certain height bound h. This means that if q is not an intersection point, then the distance $d(q, \widehat{G}) > \delta(d, h, G)$ where G is the original curve that contains $\widehat{G}$.

How do we know that the above algorithm will terminate? We must be sure that the diameters of F will be reduced by a fixed fraction $0 < f < 1$ whenever we subdivide F at the point $F(1/2)$. The constant f depends only on the degree and norm of F. The following lemma ensures this:

LEMMA 23 *Suppose $F = F[0, 1]$ is a Bezier curve and $F \to (F_0, F_1)$ where $F_0 = F[0, 1/2]$ and $F_1 = F_1[1/2, 1]$. Then* $\mathrm{diam}(F_i) \leq 2^n \mathrm{diam}(F_{1-i})$ *for* $i = 0, 1$.

Proof. Let the projection of the curves F, F_0, F_1 onto any line ℓ have signed lengths f, f_0, f_1. Thus, $|f| = \max\{|f_0|, |f_0 + f_1|, |f_1|\}$ It is enough to show that $|f_0| \leq 2^n |f_1|$. The result is trivial when $n = 1$. Suppose $n > 1$. Let the control polygon $P(F)$ be $(p_0, p_1, \ldots, p_n)$ and let the projections of the segments $[p_{i-1}, p_i]$ onto ℓ have signed lengths of d_i. Consider the control polygon $(q_0, q_1, \ldots, q_{n-1})$ where $q_i = (p_i + p_{i+1})/2$. Let this define a curve F' that splits into (F'_0, F'_1). The projection of F', F'_0, F'_1 only ℓ are the signed numbers f', f'_0, f'_1. By induction, we have $|f'_0 \leq 2^{n-1} |f'_1|$. We have $|f| = |(\sum_{i=u}^{v} d_i)|$ for some $0 \leq u \leq v \leq n$. **Q.E.D.**

10 Final Remarks and Open Problems

The present algorithm has many complex details: many details are only in the analysis, having no bearing on the implementation of the algorithm. Those details that affect the implementation might be made more palatable with two remarks: (1) Most of implementation complexity treat unlikely situations. Hence this complexity has little bearing on typical execution times. (2) Some of this implementation complexity may be inevitable if we want a complete adaptive algorithm. For instance, current nonrobust algorithms never have to worry about all the special cases because they are meaningless in an inexact setting. In any case, we pose the challenge to simplify our algorithm.

There are many open problems from our new perspective on curve intersections:

- A basic problem is the removal the condition ($C2$) of our intersection criterion. Alternatively, we want to a computational way to check this condition.
- Give a complexity analysis of our algorithm or some variant. Traditionally, not much complexity analysis has been done in subdivision algorithms.
- Implement such an algorithm and compare this to non-robust subdivision algorithms, or algebraic or partly-algebraic approaches (e.g., Wolpert's).
- Improve our separation bounds. Clearly, our current bounds overly pessimistic in reality. We already have some improvement of the bounds by exploiting the fact that they are Bezier curves. But developing novel geometric techniques for bounds would be of great interest.

Acknowledgements

This work has been carried out while the author was Visiting Professor at the School of Computer Science and Engineering at Seoul National University. Special thanks to Professor Myung-Soo Kim for his warm hospitality. This paper has greatly benefitted from discussions with Myung-Soo Kim, Gershon Elber, Joon-Kyung Seong and Iddo Hanniel. Thanks to Sung-il Pae for proofreading this. Professor Wen-ping Wang pointed out the need to distinguish between tangential crossing and tangential non-crossing cases.

References

[1] D. S. Arnon and S. McCullum. A polynomial-time algorithm for the topological type of a real algebraic curve. *J. of Symbolic Computation*, 5:213–236, 1988.

[2] C. Bajaj and M.-S. Kim. Algorithms for planar geometric models. In *Proc. 15thInt. Colloq. Automata, Languages and Programming*, pages 67–81, 1988. Lecture Notes in Computer Science.

[3] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry.* Algorithms and Computation in Mathematics. Springer, 2003.

[4] E. Berberich, A. Eigenwillig, M. Hemmer, S. Hert, K. Mehlhorn, and E. Schömer. A computational basis for conic arcs and boolean operations on conic polygons. In *Proc. 10th European Symp. on Algorithms (ESA'02)*, pages 174–186. Springer, 2002. Lecture Notes in CS, No. 2461.

[5] E. Cohen, R. Riesenfeld, and G. Elber. *Geometric Modeling with Splines: An Introduction.* A.K. Peters, Ltd, Wellesley, MA, 2001.

[6] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press, Inc, second edition, 1990.

[7] N. Geismann, M. Hemmer, and E. Schömer. Computing a 3-dimensional cell in an arrangement of quadrics: Exactly and actually! In *Proc. 17thACM Symp. on Comp. Geometry*, pages 264–273, 2001.

[8] A. J. Goldstein and R. L. Graham. A Hadamard-type bound on the coefficients of a determinant of polynomials. *SIAM Review*, 16:394–395, 1974.

[9] H. Hong. An efficient method for analyzing the topology of plane real algebraic curves. *Mathematics and Computers in Simulation*, 42:571–582, 1996.

[10] M.-S. Kim and I.-K. Lee. Gaussian approximations of objects bounded by algebraic curves. In *Proc. 1990 IEEE Int'l, Conf. on Robotics and Automation*, pages 322–326, 1990. May 13–18, Cincinnati, U.S.A.

[11] N. M. Patrikalakis and T. Maekawa. *Shape Interrogation for Computer Aided Design and Manufacturing.* Springer Verlag, Heidelberg, Germany, 2002.

[12] T. Sakkalis. The topological configuration of a real algebraic curve. *Bulletin of the Australian Math. Soc.*, 43:37–50, 1991.

[13] T. W. Sederberg and R. J. Meyers. Loop detection in surface patch intersections. *Computer Aided Geometric Design*, 5(2):161–172, 1988.

[14] R. Seidel and N. Wolpert. On the exact computation of the topology of real algebraic curves. In *Proc. 21stACM Symp. on Comp. Geometry*, pages 107–116, 2005. Pisa, Italy.

[15] R. Wein. High level filtering for arrangements of conic arcs. In *Lecture Notes in Computer Sci., vol. 2461*, pages 884–895. Springer-Verlag, 2002. Proc. 10th European Symposium on Algorithms (ESA 2002), Rome.

[16] N. Wolpert. *An Exact and Efficient Approach for Computing a Cell in an Arrangement of Quadrics.* PhD thesis, University of Saarland, Saarbruecken, Germany, Oct. 2002.

[17] N. Wolpert. Jacobi curves: Computing the exact topology of arrangements of non-singular algebraic curves. In *11th European Symposium on Algorithms (ESA)*, pages 532–543, 2003. Budapest, 15-20 September.

[18] C. K. Yap. *Fundamental Problems of Algorithmic Algebra.* Oxford University Press, 2000.