The Sorting of Points Along An Algebraic Curve

John K. Johnstone
Ph.D. Thesis

87-841
June 1987

Department of Computer Science
Cornell University
Ithaca, New York 14853-7501

THE SORTING OF POINTS ALONG AN ALGEBRAIC CURVE

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

John Keith Johnstone

August 1987

THE SORTING OF POINTS ALONG AN ALGEBRAIC CURVE

John Keith Johnstone, Ph.D.

Cornell University 1987

The area of geometric modeling is concerned with the creation of computationally efficient models of solid physical objects to facilitate their design, assembly, and analysis. In a geometric modeling system, a solid such as a robot hand or a coffee cup is modeled by a collection of points, curves, and surfaces. The sorting of points along an algebraic curve is an operation that arises frequently during the creation and manipulation of geometric models. This thesis presents a thorough investigation of sorting, including an evaluation of the two conventional methods of sorting and the presentation of a new and superior method.

A brute-force method of sorting is to trace along the curve, using Newton's method, and record the order in which the points are encountered. However, this method is inherently inefficient. A natural way to sort points along an algebraic curve is to use a rational parameterization of the curve. However, both of the main steps of this method, finding and solving a ratio-

nal parameterization of the curve, can be difficult and expensive. Indeed, many curves do not even have a rational parameterization.

We present a new method of sorting which is motivated by the observation that points on a convex segment of a curve can be sorted easily. The fundamental steps of this method are a decomposition of the curve into convex segments and a robust traversal of the curve by convex segments. This traversal is especially challenging in the neighbourhood of singularities. The points of inflection and singularities of the curve play a major role in the curve's decomposition into convex segments, which is a preprocessing step.

We analyze the complexity of the three sorting methods and then present execution times for the three methods. The new method can sort points along any algebraic curve. Moreover, for those curves that can be sorted by all three of the sorting methods, the new method is usually much more efficient than the conventional methods. We illustrate important applications of sorting, including the intersection and display of geometric models.

Biographical Sketch

John Keith Johnstone was born in Fredericton, Canada, on March 29, 1963, and entered the music program of the University of Saskatchewan in 1979. Upon graduation in 1983 with a High Honours Bachelor of Science Degree in Mathematics, he received the Governor General's Gold Medal as the most outstanding graduate of the University of Saskatchewan. In the fall of 1983, he became a graduate student in the Department of Computer Science at Cornell University on a Sage Graduate Fellowship. He was also supported throughout his four years at Cornell by an NSERC 1967 Fellowship and an Imperial Esso Graduate Fellowship. In 1986, he received his Master of Science Degree in Computer Science. In the fall of 1987, he will become an assistant professor at Johns Hopkins University.

To my parents

Acknowledgements

My advisor, John Hopcroft, was a continuous source of inspiration and a great help at focusing my work in the right direction during my initial excursions into research. I have also benefited from his unique view of robotics. I would like to thank the members of my committee: Charles Van Loan, who was also an encouraging graduate field representative and a superb teacher, and Neal Zaslaw, who helped me to satisfy my musical curiosity and made my years at Cornell much richer. It is a pleasure to acknowledge the contribution of Chanderjit Bajaj, who read parts of this thesis and made valuable comments. He clarified the use of desingularizing maps for the convex-segment method, suggested several applications of sorting, and alerted me to Collins' complexity analysis of resultant computation. I am grateful to Professors Kozen, Lichtenbaum, and Sweedler for their valuable discussions of material related to this thesis. Finally, I raise my glass to my fellow cooks and comrades at Dewitt Place.

Contents

List of Figures

Chapter 1

The Sorting of Points

1.1 Introduction

The area of geometric modeling is concerned with the creation of computationally efficient models of solid physical objects to facilitate their design, assembly, and analysis. Geometric models of physical objects are needed in many disciplines, including robotics, computer vision, computer-aided design/computer-aided manufacturing, and graphics. In a geometric modeling system, a solid such as a robot hand or a coffee cup is modeled by a collection of points, curves, and surfaces. The creation and manipulation of a solid model requires a variety of high-level operations, such as the combination of models by the boolean operations of union, intersection, and difference; the addition of blending surfaces to a model to smooth off sharp connections between edges and faces; the detection of interference between models; and the pictorial display of a model. The implementation of these high-level operations requires a number of basic tools, such as finding tangents to curves and surfaces; parameterizing lines, curves, and surfaces; measuring distances; and finding roots of equations. The development of efficient ways to perform these fundamental tasks will benefit all of the applications that use them.

The sorting of points along an algebraic curve, as evidenced by its numerous applications, is a basic tool for the manipulation of geometric models. Curve sorting has a natural definition. If S is a set of points on a curve and $\widehat{AB}$ is a segment of this curve, then to sort the points of S from A to B along $\widehat{AB}$ means to put them into the order that they would be encountered in travelling continuously from A to B along $\widehat{AB}$ (Figure 1.1). Any of the points that do not lie on $\widehat{AB}$ are ignored.

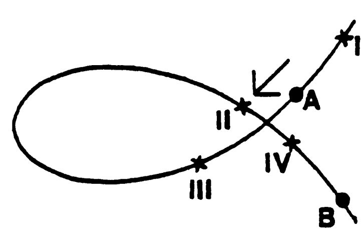


Figure 1.1: The sorted order from A to B is III, II, IV

Curve sorting has many applications in geometric modeling. We offer two examples and postpone the discussion of further applications until Chapter 5. An edge of a solid model is defined by a curve and a pair of endpoints. A point lies on the edge if and only if it lies on the curve and between the endpoints. The problem of deciding whether a point lies between two other points is a sorting problem. A more elaborate application of sorting arises in computing the intersection of two solid models. An important step of this computation is to find the segments of an edge of one model that lie in the intersection. This is done by finding and sorting the points of intersection of this edge with a face of the other model. The segments of the edge between the i^{th} and $i+1^{st}$ intersections, for i odd, are contained in the intersection of the models.

There is no serious study of curve sorting in the literature. This can be explained by the fact that, until recently, almost all of the curves in solid models were linear or quadratic, and the sorting of a curve of these low degrees is trivial (Corollary 2.1). However, as the science of geometric modeling matures and grows more ambitious, curves of degree three and higher are becoming quite common. For example, the introduction of blending surfaces into a model creates curves and surfaces of high degree. Therefore, the sorting of points along an edge of a solid model has become an important and difficult problem.

The lack of a study of curve sorting can also be explained by the presence of an obvious method for sorting points. This obvious method, which uses a parameterization of the curve, a tool with which the geometric modeler is familiar, tends to obviate a search for any other method.

This thesis will present a thorough investigation of curve sorting. We will show that there are at least three methods for sorting points along an algebraic curve. One of these methods is entirely new and particularly appealing. It will be shown to be the method of choice in many situations. The thesis is organized as follows. The next section gives a formal definition of curve sorting, and the remainder of the chapter discusses the two conventional methods of sorting. Chapters 2 and 3 present our new method. Chapter 4 is devoted to an analysis of the complexity of the new method, experimental results, and a comparison of the three methods. Applications of curve sorting, future research, and conclusions can be found in Chapter 5. Finally, there are three appendices, for definitions, technical lemmas, and a survey of parameterization algorithms.

1.2 The Sorting of Points Along a Curve

In this section, we give a formal explanation of the sorting of points along a curve. Let C be an irreducible,[1] algebraic curve described by a polynomial $f(x,y) = 0$ or the intersection of two polynomial surfaces $f_1(x,y,z) = 0$ and

[1]Terms such as these are defined in Appendix A.

$f_2(x, y, z) = 0$. (The primary representation of curves and surfaces in many solid modelers is the implicit equation. This representation is convenient for deciding if a point lies on the curve or surface, and for applying techniques from algebraic geometry, as in the creation of blending surfaces [18].) Let $\widehat{AB}$ be a segment of C, such that both A and B are nonsingular points. If $S \subset C$ is a set of nonsingular points on the curve, then (as we have said) to sort the points of S from A to B along $\widehat{AB}$ means to put the points into the order that they would be encountered in travelling continuously from A to B along $\widehat{AB}$. Points of S that do not lie on $\widehat{AB}$ are never encountered and are thus ignored. In order to avoid confusion, a vector at A is provided to indicate the direction in which the sort is to proceed from A. This is especially important when the curve is closed, since there are then two segments between A and B to choose from.

A is called the **start point**, B the **end point**, and $\widehat{AB}$ is the **sort segment**. The points that are to be sorted (the points of S) are called the **sortpoints**. We shall often refer to the sorting of points along a curve as **curve sorting**.

The sorting of points along a curve is more sophisticated than the sorting of numbers or names. In particular, start and end points are necessary, and the sorted set is often a strict subset of the unsorted set. These changes are necessary in order to resolve the ambiguity of sorting on a closed curve, where order is cyclic. The changes are also useful for geometric modeling applications, since only a part of the curve (viz., the edge) is of interest.

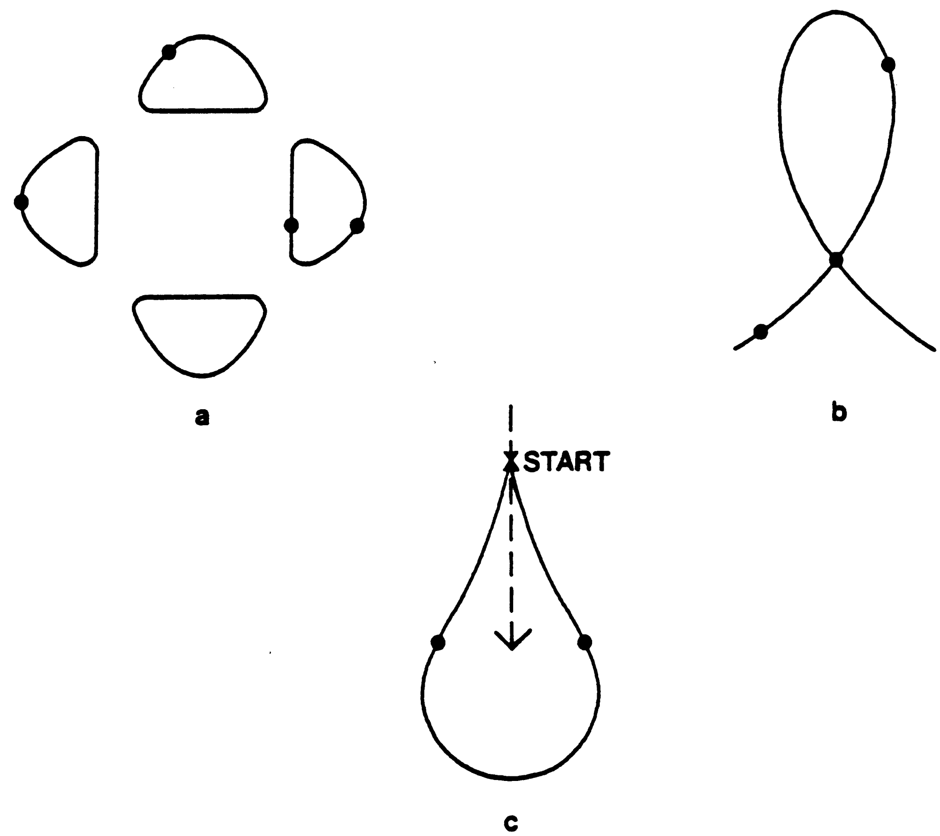


Figure 1.2: Some sorts are ambiguous

Every curve in this thesis can be assumed to be both algebraic and irreducible. We also assume that all of the sortpoints of a sorting problem lie on the connected component that contains the sort segment, since the order of a set of points is unclear if the points lie on different connected components (Figure 1.2(a) and Section 5.2.2). Finally, we assume that the start point, the end point, and the sortpoints are nonsingular points, because the order of a set of points that includes singularities can be ambiguous (Figure 1.2(b-c)).

1.3 The Parameterization Method

A natural way to sort points along a curve is to use a rational parameterization of the curve (i.e., a parameterization $(x(t),y(t))$ or $(x(t),y(t),z(t))$ such that $x(t)$, $y(t)$, and $z(t)$ can each be expressed as the quotient of two polynomials in t). The parameter values t_i of the points (x_i, y_i) are computed and sorted by increasing t_i values. The points that occur before the start point or after the end point of the sort segment are discarded.

Example 1.3.1 *Consider the curve* $x^3 + xy^2 + 16x^2 - 4y^2$ *(a pedal of a parabola) with start point* $A = (1, \sqrt{\frac{17}{3}})$*, end point* $B = (3, -3\sqrt{19})$*, sortpoints* $I = (3, 3\sqrt{19})$*,* $II = (-1, \sqrt{3})$*,* $III = (-2, -2\sqrt{\frac{7}{3}})$*, and* $IV = (1, -\sqrt{\frac{17}{3}})$*, and direction* V_A *from* A*, as shown in Figure 1.1. A parameterization of the curve is* $x = \frac{4t^2-16}{t^2+1}$*,* $y = \frac{4t^3-16t}{t^2+1}$*. We compute a point* d *on the curve close to* A *in the direction* V_A*.*[2] *The parameter value of* d *indicates the order (increasing or decreasing) in which the parameter values should be sorted: the values are sorted in increasing order if and only if the parameter value of* d *is greater than that of* A*. For this example,* $d = (0.9, \sqrt{\frac{13.689}{3.1}})$.

[2] d is found by crawling from A, a technique that is described in Section 1.4.

The parameter value associated with $I = (3, 3\sqrt{19})$ *is determined by solving the system* $\{3 = \frac{4t^2-16}{t^2+1}, 3\sqrt{19} = \frac{4t^3-16t}{t^2+1}\}$*, yielding* $t = \sqrt{19}$*. The parameter values [II ,* $t = -\sqrt{3}$*], [III ,* $t = \sqrt{\frac{7}{3}}$*], [IV ,* $t = -\sqrt{\frac{17}{3}}$*], [A ,* $t = \sqrt{\frac{17}{3}}$*], [B ,* $t = -\sqrt{19}$*], and [d ,* $t = \sqrt{\frac{16.9}{3.1}}$*] are computed in an analogous fashion. After sorting by parameter values, the point list becomes B, IV, II, III, d, A, I. Sorting from A to B in the appropriate direction yields the desired sorted list III, II, IV.*

There are three reasons that we insist that the curve's parameterization be rational. Consider a parameterization that involves n^{th} roots, such as $x(t) = \sqrt{t} + 2\sqrt[3]{t}$. In solving such a parameterization for the parameter value of a given point, it can be unclear which n^{th} root to use. Another inconvenient property of non-rational parameterizations is that their representation is difficult. They cannot be represented by the coefficients of the parameterization: they must be represented symbolically (as in a computer algebra system like MACSYMA), which is less efficient. For example, a parameterization of the devil's curve $y^4 - x^4 - y^2 + 4x^2 = 0$ is

$$x = \cos(t)\sqrt{\frac{sin^2(t) - 4cos^2(t)}{sin^2(t) - cos^2(t)}},\ y = \sin(t)\sqrt{\frac{sin^2(t) - 4cos^2(t)}{sin^2(t) - cos^2(t)}}$$

Finally, there is no algorithm for the automatic parameterization of curves that do not have a rational parameterization. Thus, there are problems with all three aspects of a non-rational parameterization: computation, representation, and solution.

1.3.1 Parameterization

Definition 1.1 *A curve is* **rational** *if it has a rational parameterization.*

The translation of an implicit representation of a curve into a parametric representation, which is one of the key steps of the parameterization method of sorting, has received some attention in the literature. There are constructive methods for the parameterization of plane curves of low order (viz., two and three) [2,3,19] and rational plane curves [4]. There are also constructive methods for the parameterization of surfaces of low degree (again, two and three) [2,3,28], but not of high degree even if the surface has a rational parameterization. (The parameterization of surfaces is important because, as shown below, a space curve's parameterization can be developed from a parameterization of one of the surfaces that defines the space curve.) Appendix C presents various methods for the parameterization of low degree curves and surfaces.

A rational plane curve of order n can be parameterized by establishing a one-to-one correspondence between the points of the curve and a one-parameter family of curves of degree max(n − 2,1) through well-chosen single and double points of the curve [4].

Example 1.3.2 *Consider the parameterization of the circle* $x^2+y^2-1=0$. *Let* P *be a point of the circle (say* $P=(-1,0)$*) and let* L_t *be the line through* P *of slope* t. *There is a one-to-one correspondence between the* *lines through P and the points of the circle, which can be used to construct a parameterization: the parameter value of a point Q is the slope of the line through P that hits Q. The equation of L_t is $y = tx + t$. The point of the circle associated with L_t satisfies $y = tx + t$ and $x^2 + y^2 - 1 = 0$, so $x^2 + (tx + t)^2 - 1 = 0$. Using the quadratic formula, $x = -1$ or $\frac{1-t^2}{1+t^2}$. The latter root is clearly the one of interest. Therefore,*

$$y = tx + t = t(\frac{1-t^2}{1+t^2}) + t = \frac{2t}{1+t^2}$$

and a parameterization of the circle is $(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2})$.

A space curve's parameterization can sometimes be derived from a parameterization of one of its two constituent surfaces [20,29]. We illustrate the method with an example.

Example 1.3.3 *Consider the space curve $S \cap T$, where $S \equiv x^2 + y^2 - 1 = 0$ and $T \equiv 2x^2 + y^2 - z^2 = 0$ are surfaces. We find a parameterization for S, $\{x = \frac{1-t^2}{1+t^2}, y = \frac{2t}{1+t^2}, z = s\}$, and substitute these parametric equations into the implicit equation for T, yielding $\frac{2(1-t^2)^2+4t^2}{(1+t^2)^2} - s^2 = 0$. We then solve for s in terms of t: $s = \frac{\sqrt{2t^4+2}}{1+t^2}$. We conclude that a parameterization for the intersection of the two surfaces is $\{x = \frac{1-t^2}{1+t^2}, y = \frac{2t}{1+t^2}, z = \frac{\sqrt{2t^4+2}}{1+t^2}\}$. Notice that this parameterization is not appropriate for sorting, because it is not rational.*

Unfortunately, it may be hard or impossible to find a parameterization for either of the surfaces, and it may be impossible to solve for s in terms of t (and vice versa) after substituting the parametric equations of one surface into the implicit equation of the other. In particular, problems will arise when the degree of s is five or more, since there is no general formula for the solution of equations of degree greater than four [16]. Therefore, this technique of space curve parameterization is quite restricted.

We conclude that plane curves of low degree, rational plane curves, and a restricted class of rational space curves can be automatically parameterized.

1.3.2 Weaknesses of the Parameterization Method

The two main steps of the parameterization method of sorting are to find a rational parameterization of the curve and to find the parameter values of the sortpoints by solving this parameterization. We shall show that there are problems with both of these steps.

Only a strict subclass of algebraic curves are rational, and the proportion of algebraic curves that are rational decreases as the order of the curve increases. Therefore, there are many curves that cannot be sorted by the parameterization method, simply because they do not have a rational parameterization. These facts are established by considering the genus of a curve.

Definition 1.2 *The* **genus** *of an irreducible, algebraic, plane curve is*

$$\frac{(n-1)(n-2)}{2} - \sum \frac{r_i(r_i-1)}{2}$$

where n is the order of the curve, the sum is over the singularities P_i of the curve,[3] *and r_i is the multiplicity of the curve at the singularity P_i. The genus is nonnegative, and it is zero if and only if the curve has the maximum number of singularities allowed for a curve of its order.*

Theorem 1.1 ([35, p. 180]) *A plane curve is rational if and only if its genus is 0.*

Example 1.3.4 *It can be shown that any irreducible quartic (degree four) space curve that is generated by the intersection of two degree two surfaces is non-rational.*

The genus of any quartic plane curve with an ordinary double point is 1, so no curve of this type has a rational parameterization. For example, the lemniscate of Bernoulli ($x^4+y^4+2x^2y^2-4x^2+4y^2=0$) is such a curve, and its parameterization is $x = \frac{2cos(t)}{1+sin^2(t)}$, $y = \frac{2sin(t)cos(t)}{1+sin^2(t)}$, $-\pi \leq t \leq +\pi$ [27].

Even if a rational parameterization for the curve can be found, it is still necessary to solve for the parameter values associated with the set

[3] Singularities must be counted properly and singularities at infinity must be included. See [35, pp. 80-84].

of points that are to be sorted. This is another weakness of the parameterization method, because the solution of a polynomial of high degree is usually expensive. Even for the tame example of the pedal of a parabola (Example 1.3.1), which has a parameterization involving degree three polynomials, each solution for the parameter value of a sortpoint consumes on the order of 120 milliseconds.[4] We shall see in Chapter 4 that this is a nontrivial expense. We conclude that the parameterization method of sorting is limited and slow.

There are two fundamental ways of representing a curve: the implicit equation and the parameterization. The difficulties that arise with the parameterization method of sorting reflect the difficulty of working with the parameterization representation of a curve in an environment where the implicit equation is the original representation.

1.4 The Crawling Method

Another method of sorting points along a curve is the crawling method. This is a brute-force method that sorts a set of points by tracing along the sort segment and recording the order in which the sortpoints are encountered during this trace. The curve is traced by making small jumps along

[4]Using Common Lisp on a Symbolics Lisp Machine, and the ZEROIN algorithm of Dekker and Brent for solving nonlinear equations [13].

it, using Newton's method. For example, consider a small jump of size ϵ from the point $P = (x_0, y_0)$ of the curve $f(x, y) = 0$. Depending upon the behaviour of the curve in the neighbourhood of P, x_0 or y_0 is incremented or decremented by ϵ. Suppose that x_0 is incremented, effectively jumping off of the curve to $(x_0 + \epsilon\ ,\ y_0)$. A root y' of $f(x_0 + \epsilon, y)$ is found by applying Newton's method, with initial guess y_0. $(x_0 + \epsilon, y')$ is a point of the curve that lies close to (x_0, y_0), but it is a step further along the curve from the original point.

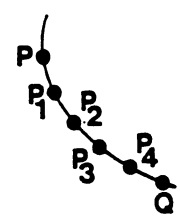


Figure 1.3: Tracing a curve from P to Q

Progress is made along the curve by these small jumps. For example, in Figure 1.3, a tracing of the curve from P to Q may involve jumping to P_1, P_2, P_3, and P_4. The details of tracing along a curve, including a discussion of how to trace robustly through a singularity, are presented in [9].

If a jump is made to a point within some ball of radius δ about a sortpoint x, then we assume that the trace has reached x and we insert x into the sorted list that is being accumulated. If δ is small, then this is a reasonable assumption. However, this assumption is not entirely robust (Figure 1.4).

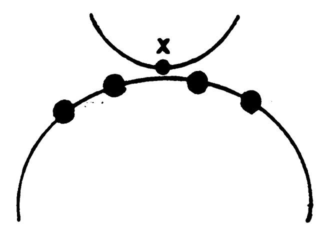


Figure 1.4: x may be sorted improperly by the crawling method

The major weakness of the crawling method is that it must make small jumps, in order to ensure that the crawl proceeds smoothly along the curve and does not get confused. If ϵ is large, then it is possible to jump discontinuously to another part of the curve (Figure 1.5(a)), or even to jump completely off of the end of the curve (Figure 1.5(b)). The jumps must also be small in order to avoid jumping over two or more sortpoints at the same time, which would cause problems in sorting, and to avoid ignoring a sortpoint x by tracing through it without jumping into the δ-ball about x that triggers x's insertion into the sort. As a result, the crawling method is very slow unless the sort segment is short.

Another undesirable property of the crawling method is that its speed depends upon the length of the sort segment rather than upon the number of points to be sorted. The crawling method does have the advantage

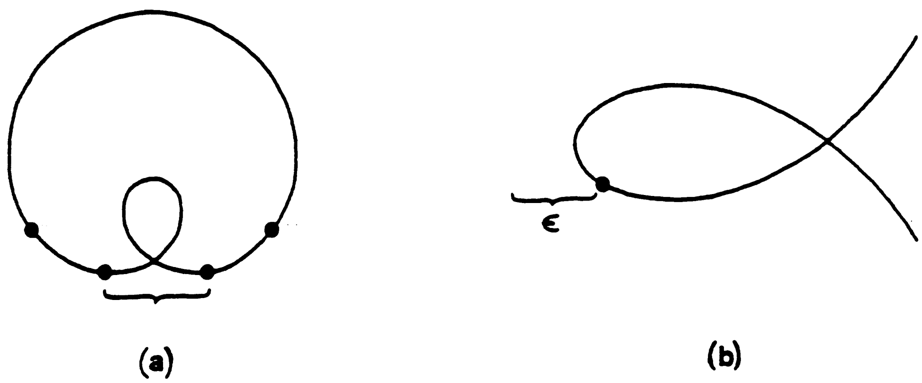


Figure 1.5: Jumps must be small

that it does not require any preprocessing, such as the computation of a parameterization.

The weaknesses that we have observed in the parameterization and crawling methods of sorting points along a curve suggest that another method is necessary: one that will perform more efficiently on a wider selection of algebraic curves. The next chapter presents such a method.

Chapter 2

The Convex-Segment Method of Sorting

2.1 Sorting a Convex Segment

The observation that motivates the new method is that a convex segment can be sorted easily. Since every curve is a concatenation of convex segments (Figure 2.1), this suggests a divide and conquer strategy.

The new method only applies directly to plane curves. However, the sort of a space curve (i.e., a curve that does not lie in a plane) can be mapped into a sort of a plane curve, as we shall see in Section 2.6, so there is no loss of generality.

Definition 2.1 *A segment* $\widehat{PQ}$ *of a plane curve is* **convex** *if no line has*

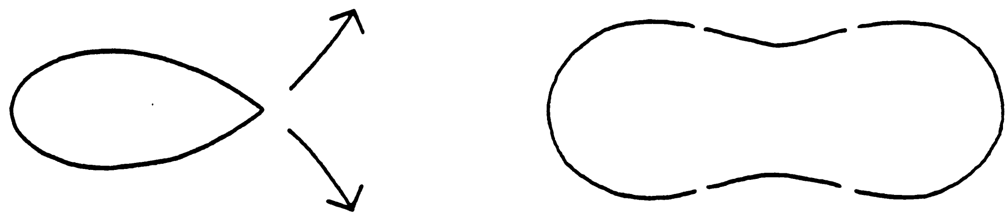


Figure 2.1: A curve is a collection of convex segments

more than two intersections with $\overset{\frown}{PQ}$*, unless all of the intersections occur at the same point and are even in number*[1] *(Figure 2.2).*

There is another useful characterization of convexity: a plane segment is convex if it lies entirely on one side of the closed halfplane determined by the tangent line at any point of the segment [12].

Definition 2.2 *A polygon* P *is* **convex** *if, for all* $A, B \in P$*, the line segment* $\overline{AB}$ *is entirely contained in* P*. The* **convex hull** *of a set of points* S *is the smallest convex polygon that contains* S*.*

Notation 2.1 *Let* P *and* Q *be points of a curve* C*.* $\overset{\frown}{PQ}$ *is the segment of the curve joining* P *to* Q*.*[2]

The following theorem establishes that it is simple to sort a set of points on a convex segment.

[1]The normal definition of convexity is that $\overset{\frown}{PQ}$ is convex if no line has more than two intersections with $\overset{\frown}{PQ}$. We amend this definition because, for our purposes, a curve such as $y = x^4$ can be considered convex. Thus, in the terminology of Section 2.2, we allow a convex segment to contain flexes of even order.

[2]If the curve is closed, then the context should make clear which segment is intended.

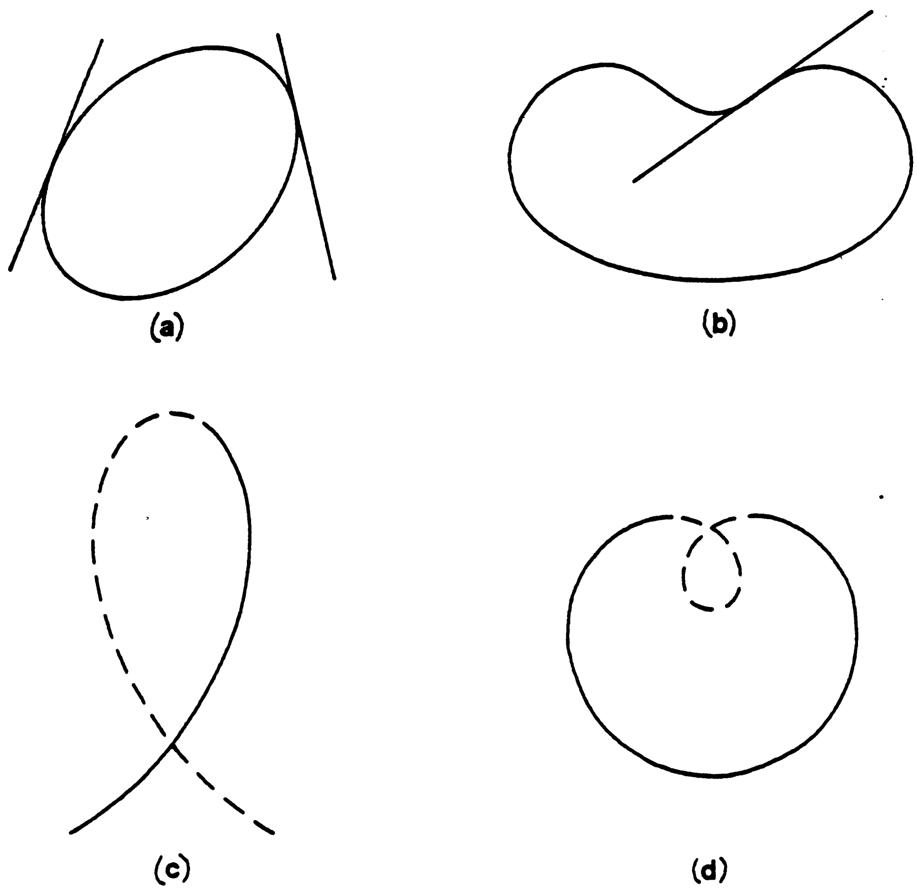


Figure 2.2: (a) is convex; (b)-(d) are not

Theorem 2.1 *Let $p_1, \ldots, p_n$ be points of a convex segment $\widehat{AB}$, and let H be the convex hull of A, B, $p_1, \ldots, p_n$. Then the vertices of H are A, B, $p_1, \ldots, p_n$. Moreover, the order of the vertices on the boundary of H is exactly the order of the points on $\widehat{AB}$ (Figure 2.3).*

Proof It is sufficient to show that the polygon created by joining the points A, B, $p_1, \ldots, p_n$ in sorted order is their convex hull. Let this polygon

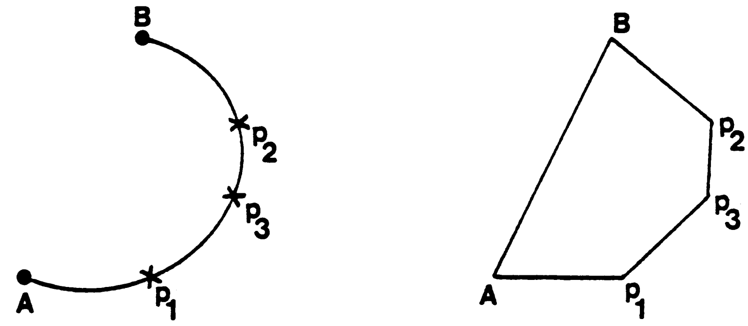


Figure 2.3: An example of Theorem 2.1

be $P = r_0 r_1 \ldots r_{n+1}$, where $r_0 = A$, $r_{n+1} = B$, and $r_1, \ldots, r_n$ is the sorted order of the p_i on the curve. Let $E = \overline{r_i r_{i\oplus 1}}$ be an edge of P (where $\oplus$ is addition mod $n + 2$), $\overleftrightarrow{E}$ the infinite line containing E, and $\widehat{E} = \widehat{r_i r_{i\oplus 1}}$. By convexity, $\overleftrightarrow{E}$ can have only two intersections with $\widehat{AB}$, those at r_i and $r_{i\oplus 1}$. Therefore, all of $\widehat{E}$ must lie on the same side of $\overleftrightarrow{E}$, with the rest of $\widehat{AB}$ on the other side.[3] Since none of the r_j lie in between r_i and $r_{i\oplus 1}$, all of the points r_j lie on one side of the halfplane defined by the edge E, and only the endpoints of E lie on $\overleftrightarrow{E}$. Suppose, for the sake of contradiction, that the continuation $\overleftrightarrow{E}$ of E intersects another edge $\overline{r_j r_{j\oplus 1}}$. Either r_j and $r_{j\oplus 1}$ lie on opposite sides of $\overleftrightarrow{E}$ or one of $r_j, r_{j\oplus 1}$ lies on $\overleftrightarrow{E}$, a contradiction in either case. Therefore, the continuation of any edge of the polygon P does not strike any other edge of P. By Lemma B.3, P is convex. Since a

[3]A point of tangency to $\overleftrightarrow{E}$ counts as two intersections, so $\widehat{AB}$ cannot stay on $\widehat{E}$'s side by being tangent at r_i or $r_{i\oplus 1}$.

convex polygon is the convex hull of its vertices, P is the convex hull of the r_i's. ▮

Corollary 2.1 *Conics can be sorted easily.*

Proof Conics are convex, since a line can have only two intersections with an irreducible curve of order two (Theorem B.1). ▮

When Theorem 2.1 is used to sort a set of points on a convex segment, there is no need to actually create the convex hull. Let $Q = \frac{1}{3}(A + B + p_1)$, the barycenter of A, B, and p_1. Q will lie in the interior of the convex hull of A, B, p_1, ..., p_n (Figure 2.4). Consider the angles that the lines $\overleftrightarrow{QA}$, $\overleftrightarrow{QB}$, $\overleftrightarrow{Qp_1}$, ..., $\overleftrightarrow{Qp_n}$ make with the positive x-axis, and sort these angles from $\angle\overleftrightarrow{QA}$ to $\angle\overleftrightarrow{QB}$ (where $0 \equiv 2\pi$). Since A, B, p_1, ..., p_n are the vertices of a convex polygon and Q is a point in the interior of this polygon, the order of the angles from $\angle\overleftrightarrow{QA}$ to $\angle\overleftrightarrow{QB}$ is equivalent to the order of the vertices on the polygon from A to B. Therefore, p_1, ..., p_n can be sorted on $\widehat{AB}$ by sorting the angles that they make with a central point.

Once it is realized that points on a convex segment can be sorted easily, the focus on curve sorting can change, since the problem has essentially been reduced from sorting points to sorting convex segments. Of course, the curve must first be divided up into convex segments. We consider this division problem in the next section.

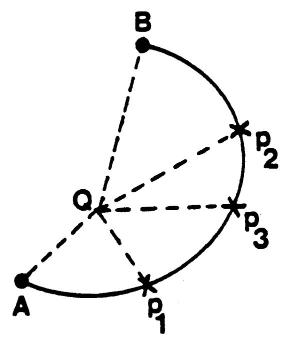


Figure 2.4: Sorting points on a convex segment by sorting angles

2.2 Convex Segmentation

The crucial step of the new method is the decomposition of the curve into convex segments. This decomposition allows us to take advantage of the simplicity of sorting points on a convex segment. The decomposition is achieved by the tangents at certain special points of the curve: the singularities (Appendix A) and flexes.

Definition 2.3 *A* **point of inflection**, *or* **flex** *for short, of a curve F is a nonsingular point $P \in F$ whose tangent has three or more intersections with F at P. (The tangent of most simple points intersects the curve twice at the point of tangency.) The number of intersections of the flex's tangent with the curve at the flex is called the* **order** *of the flex. A flex of odd order is called a* **flox**.[4]

[4]This is our own term. It is not used in the literature.

Example 2.2.1 *The origin of $y = x^3$ is a flox.*

The interesting property of a flex is that the curve can only change its direction of curvature (from convex to concave or vice versa) at a flex or a singularity. In other words, flexes are the only nonsingular points P such that the curve in any neighbourhood of P can lie on both sides of P's tangent. This is an important property because the curve in the neighbourhood of a point P of a convex segment always lies on one side of P's tangent. Since the curve in the neighbourhood of a flex of even order does not lie on both sides of P's tangent (Lemma B.4), we will only be interested in floxes.

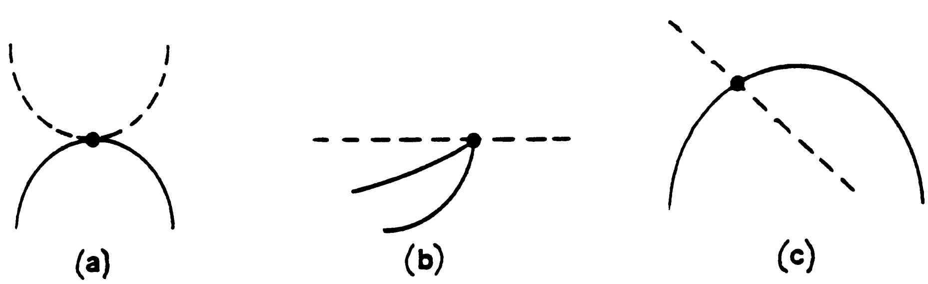


Figure 2.5: (a-b) touch (c) touch and cross

Definition 2.4 *The* **inside** *(resp.,* **outside***) of a plane curve f(x,y) = 0 is the halfplane f(x,y) < 0 (resp., f(x,y) > 0). Two curves* **touch** *at P if they intersect at P. A line L* **crosses** *a plane curve C at P if L touches C at P and L lies on both the inside and outside of C in any neighbourhood of P (Figure 2.5).*

The tangents at the singularities and floxes of a curve subdivide the plane of the curve into several cells and split the curve into several segments. The following theorem establishes that each of these segments is convex.

Theorem 2.2 *Let T be the set of tangents of the singularities and floxes of a curve F, and let $\widehat{PQ}$ be a nonconvex segment of F. Then some tangent of T touches or crosses $\widehat{PQ}$. (Moreover, unless $\widehat{PQ}$ contains a singularity, some tangent of T crosses $\widehat{PQ}$.)*

Proof Assume, without loss of generality, that $\widehat{PQ}$ does not contain a flox or a singularity. (If it does, then we are done.) By Lemma B.6, there exists a line L that crosses $\widehat{PQ}$ at three (or more) distinct points. Let x_1, x_2, and x_3 be three of these points such that $x_2 \in x_1\widehat{\ }x_3$ and $x_1\widehat{\ }x_3 \cap L = \{x_1, x_2, x_3\}$ (i.e., x_1, x_2, and x_3 are as close together as possible on $\widehat{PQ}$). $x_1\widehat{\ }x_3$ does not change its direction of curvature, since there is no flox or singularity on $\widehat{PQ}$. Moreover, $x_1\widehat{\ }x_3$ is not a line segment, otherwise $x_1\overleftrightarrow{\ }x_3$ would be a component of the curve (Theorem B.1), contradicting the irreducibility of the curve. Therefore, without loss of generality, we can assume that $x_1\widehat{\ }x_3$ looks like Figure 2.6(a). Let R be the closed region bounded by $x_1\widehat{\ }x_3$ and $\overline{x_1x_3}$ (Figure 2.6(b)). We will show that R contains a flox or a singularity. This will complete the proof, since the tangent of a point inside R must cross $x_1\widehat{\ }x_3 \subset \widehat{PQ}$ at least once. (The tangent must cross the boundary of R twice, and at most one of these intersections can be with $\overline{x_1x_3}$.)

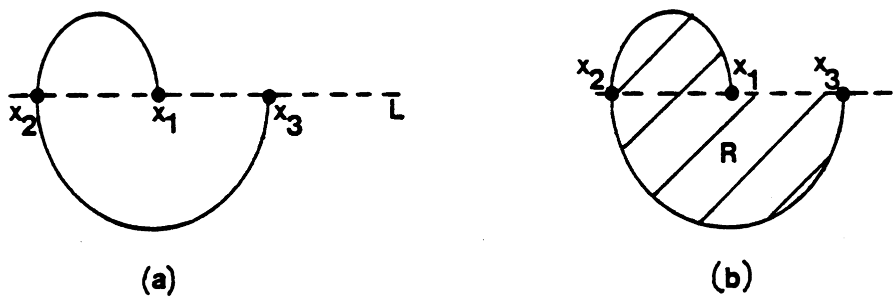


Figure 2.6: (a) $x_1\widehat{x}_3$ (b) the region R

The curve lies inside of R as it leaves $x_1\widehat{x}_3$ from x_1 and outside of R as it leaves $x_1\widehat{x}_3$ from x_3. Therefore, the curve must cross the boundary of R after it leaves $x_1\widehat{x}_3$ from x_1, either because it must join with x_3 (if the curve is closed) or because an infinite segment of an algebraic curve cannot remain within a closed region (if the curve is open, using Lemma B.5). The curve cannot cross the $x_1\widehat{x}_3$ boundary of R, since $x_1\widehat{x}_3 \subset \widehat{PQ}$ is nonsingular by assumption. Therefore, the curve must cross $\overline{x_1x_3}$ after it leaves $x_1\widehat{x}_3$ from x_1.

As the curve leaves $x_1\widehat{x}_3$ from x_1, it lies on the opposite side of x_1's tangent from $\overline{x_1x_3}$. Therefore, after the curve leaves $x_1\widehat{x}_3$ from x_1 and before it leaves R, the curve must cross x_1's tangent inside of R, in order to reach $\overline{x_1x_3}$. In order to cross over x_1's tangent, the curve must cross itself or change its curvature inside of R (Figure 2.7), otherwise it will spiral around inside R forever. Therefore, R contains a singularity or a flox. ∎

Figure 2.7: The curve must cross itself or change its curvature in travelling from x_1 to $\overline{x_1 x_3}$

Corollary 2.2 *The tangents of the singularities and floxes of a curve divide the curve into convex segments.*

Example 2.2.2 *Figure 2.8 illustrates the division of a curve into convex segments by the tangents at singularities and floxes.*

Theorem 2.2 establishes that the segmentation of a curve by its singularities and the points where the curve crosses a singularity/flox tangent will be a convex segmentation. Therefore, the nonsingular points at which the curve touches (but does not cross) a singularity/flox tangent can be ignored. For example, the convex segment from W_1 in Figure 2.9 should be $\widehat{W_1 W_3}$ rather than $\widehat{W_1 W_2}$.

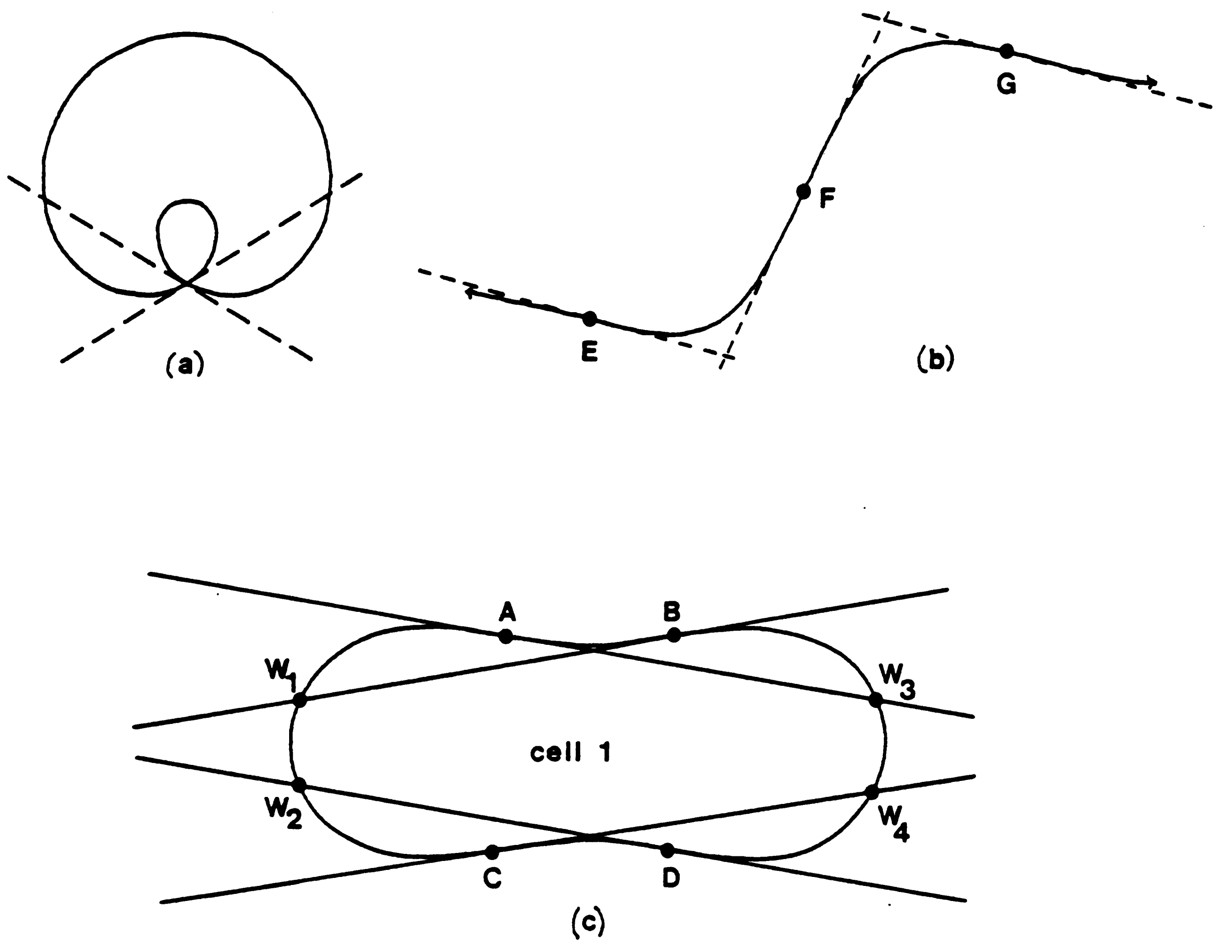


Figure 2.8: Convex segmentation of (a) limacon of Pascal (b) serpentine (c) Cassinian oval

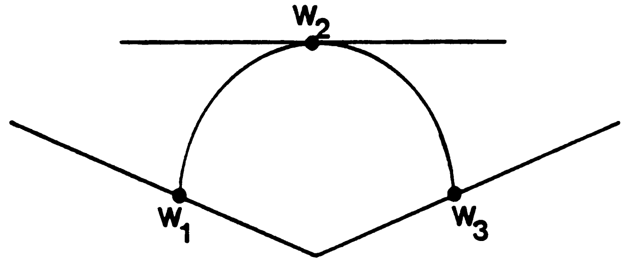


Figure 2.9: Nonsingular points where the curve merely touches a tangent are ignored

2.3 The Cell Partition and the Sorting of Convex Segments

2.3.1 The Cell Partition

Consider a subdivision of the plane into cells by the tangents at the singularities and floxes of a curve. Some of the cells contain a convex segment of the curve, some of the cells contain several convex segments of the curve, and the rest of the cells do not contain any of the curve (Figure 2.8). This subsection develops some terminology to describe this situation.

Definition 2.5 *A* **cell partition** *of a plane curve F is a partition of the plane into convex polygons, called* **cells,** *by the tangents at the singularities and the floxes of F. A line segment that forms part of the boundary of a cell is called a* **cell segment**. *A tangent of a singularity or flox of F is called a* **wall** *of the cell partition. Consider the points where the curve intersects* *the cell partition. Many of these intersections arise intentionally: i.e., the intersection is a singularity or a flox through which a wall of the cell partition was intentionally passed. The other intersections arise indirectly and are called* **incidental** *curve points. The points of intersection of the curve with the cell partition are collectively called* **curve points.**[5]

The curve is decomposed into convex segments by the cell partition. A **multisegment cell** *is a cell that contains more than one convex segment. The endpoints of each convex segment are curve points. If* $\widehat{VW}$ *is a convex segment of the partition, then V is called a* **partner** *of W (and vice versa). A curve point separates two convex segments and thus usually has two partners. However, a curve point might only have one partner: a convex segment that goes off to infinity within a cell C will have only one endpoint, and this curve point will not have a partner with respect to C.*

Example 2.3.1 *The cell partition of the serpentine (Figure 2.8(b)) has three walls and three curve points, each of which is a flox. E and F are partners, as are F and G. The cell partition of the Cassinian oval (Figure 2.8(c)) has four walls, arising from the four flexes A, B, C, and D. There are four incidental curve points:* W_1, W_2, W_3, W_4. *Cell 1 is the only multisegment cell.* W_1 *and* W_2 *are partners in cell 1, as are* W_3 *and* W_4.

[5] Recall that the nonsingular points of intersection where the curve only touches a wall are ignored.

A cell partition is not simply a collection of walls. It is a large data structure that defines interrelationships between cells, walls, curve points, and convex segments. It contains information that is needed in the implementation of the new sorting method, such as the walls of each cell (in implicit and parametric representations); for each of these walls, the side that the cell lies on; the curve points on each cell segment; and adjacency information, such as the two cells that border a cell segment.

The creation of the cell partition of a curve is a preprocessing step that is entirely independent of the sorting of any points on the curve.

2.3.2 The Sorting of Convex Segments

We complete our description of the new sorting method by discussing how to determine the order of the convex segments on the curve. The sorting of the convex segments is done in an unusual manner. In fact, to say that we 'sort' the convex segments is a misnomer. Unlike a normal sort, we do not create a list of convex segments and rearrange them into proper order. Instead, the points from A to B are sorted by traversing the curve from A to B by convex segments, stepping from curve point to curve point. The next convex segment is determined only when we need to move to it.[6] As

[6]This technique is reminiscent of lazy evaluation in compiler theory. Since we sort the convex segments 'lazily', only those convex segments that lie on the sort segment are encountered and sorted.

each convex segment is encountered, the points that lie on it are found and sorted (using Theorem 2.1), and this subsort is appended to the end of a global sort which is being accumulated. Thus, the sorting of the convex segments is interleaved with the sorting of the sortpoints.

A crucial step in the traversal of a curve by convex segments is the determination of the next convex segment. Given a convex segment $\overset{\frown}{VW}$, we must be able to find the convex segment that follows $\overset{\frown}{VW}$ from W. Suppose that $\overset{\frown}{VW}$ lies in cell C and W lies on the boundary of cells C and D. The problem of finding the convex segment that follows $\overset{\frown}{VW}$ from W reduces to finding the partner of W in D. If the cell D contains only one convex segment (as is often the case), then this is trivial. Chapter 3 discusses how to determine the partner of a curve point in a multisegment cell.

The other main step in the traversal of a curve by convex segments is the computation of the points that lie on a given convex segment. It turns out that the theory that must be developed to solve this problem is the same as for the solution of the above next-convex-segment problem. Chapter 3 discusses how to find the convex segment that contains a given point of the curve and, thus, how to find the points that lie on a given convex segment. The first step is to find the cell that contains the point: a point lies in a cell if and only if it lies on the proper side of each of the walls that define the cell. If the point's cell contains only one convex segment, then this convex segment contains the point. However, if the cell contains several convex segments, then the decision is much more complicated.

The first convex segment is determined by first finding the convex segment $\widehat{VW}$ that contains the start point S. Recall that, as part of the definition of a sorting problem, a vector at S is provided to indicate the direction in which the sort is to proceed from S. This vector can be used to determine whether the first convex segment is $\widehat{SV}$ or $\widehat{SW}$: $\widehat{SV}$ is the first convex segment if and only if the vector points to the inside of the chord $\overleftrightarrow{SV}$ (see Appendix A's definition).

We have now presented the fundamentals of our new method of curve sorting. We refer to it as the convex-segment method of sorting.

Example 2.3.2 *Consider the sorting of points along a Cassinian oval (Figure 2.10). We determine that the startpoint S lies on $\widehat{W_2C}$ and use the vector at S to choose the subsegment $\widehat{W_2S}$ as the first convex segment. There are no points on $\widehat{W_2S}$, so we move on. The next convex segment is $\widehat{W_1W_2}$, since W_1 is W_2's partner in cell 1. There are two sortpoints in cell 1 (P_1 and P_5), but only P_1 lies on $\widehat{W_1W_2}$. We make P_1 the first element of the sort. We jump to the next convex segment $\widehat{W_1A}$ and sort the two points, P_2 and P_3, that lie on this convex segment by sorting the angles that W_1, P_2, P_3, and A make with a central point. We add P_2 and P_3 to the global* *sort, and move on to the next convex segment* $\widehat{AB}$. *We immediately move on to* $\widehat{BW_3}$, *since we find that* $\widehat{AB}$ *does not contain any sortpoints. Both END and* P_4 *lie on* $\widehat{BW_3}$. *The presence of END indicates that this is the last convex segment that needs to be considered. Upon sorting END and* P_4, P_4 *is discarded because it comes after END. The final sorted list is* P_1, P_2, P_3.

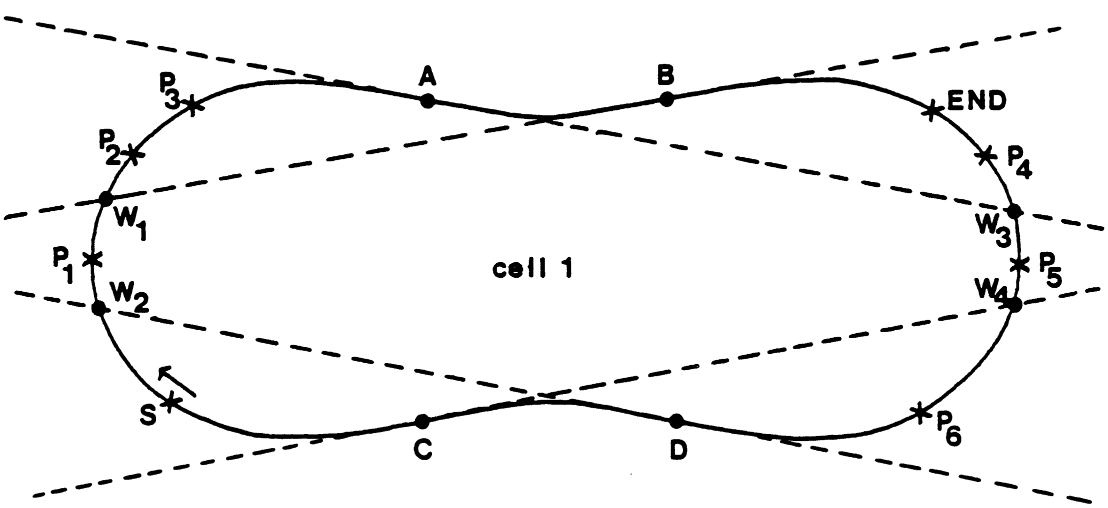


Figure 2.10: Sorting a Cassinian oval

2.4 Resolving the Ambiguity at Singularities

The traversal of a curve by convex segments is especially challenging in the neighbourhood of a singularity. In particular, it can be ambiguous which branch of the curve should be followed from a singular curve point.

Example 2.4.1 *Consider the curve in the neighbourhood of A on Figure 2.11(a). It is ambiguous whether this is two semicircles touching or two flexes crossing. In particular, it is not clear whether P_2 or P_3 follows P_1.*

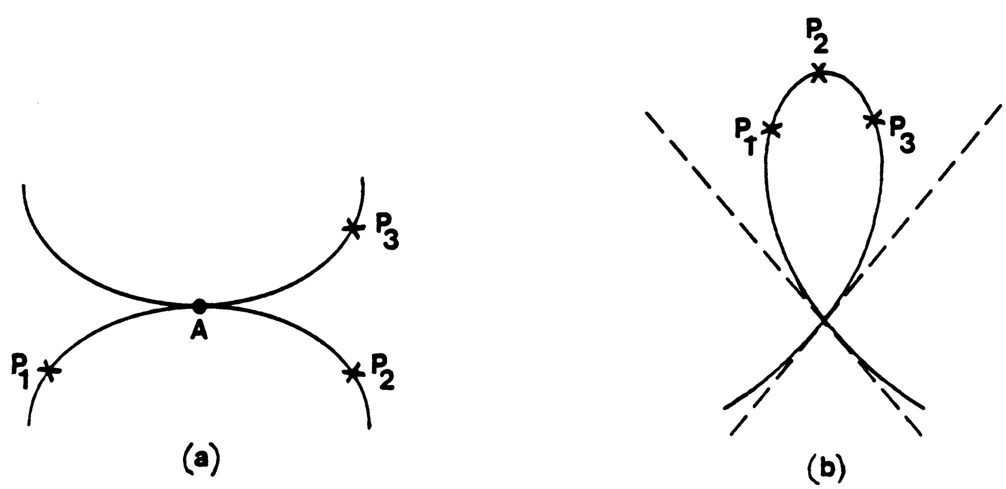


Figure 2.11: Ambiguity about a singularity

Consider the sorting of points on a loop around an ordinary singularity (Figure 2.11(b)). It is not immediately clear whether the order is P_1, P_2, P_3

or P_3, P_2, P_1.

This problem is resolved by finding, for each branch that passes through a singularity, a pair of points, one on either side of the singularity. These two points serve to guide the sort through the singularity along the proper branch. Before we discuss how to find these points, we offer an example of how they are used.

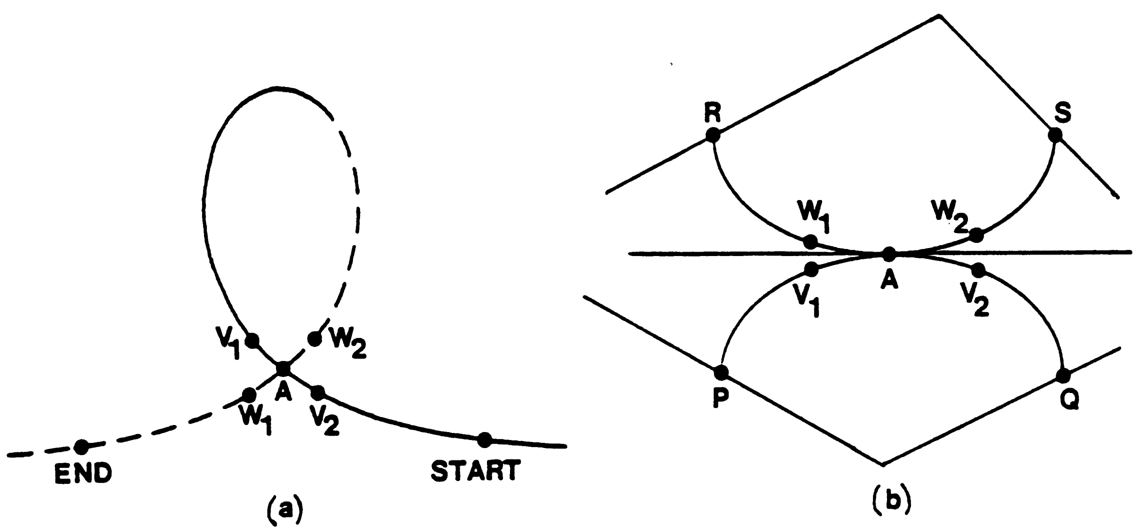


Figure 2.12: Resolving the ambiguity at a singularity

Example 2.4.2 *We associate four points with the singularity* A *of Figure 2.12(a):* V_1, V_2, W_1, *and* W_2. V_1 *is paired with* V_2 *on the solid arc, and* W_1 *is paired with* W_2 *on the dotted arc. When this curve is sorted, rather than traversing the curve from* $START$ *to* A, A *to* A, *and* A *to* END, *the traversal proceeds from* $START$ *to* V_2, V_1 *to* W_2, *and* W_1 *to* END. *Notice* *that if the traversal reaches a point associated with a singularity such as V_2, then the traversal continues from V_2's partner, not V_2 itself.*

Suppose that we have determined that the curve of Figure 2.12(b) is actually two semicircles touching at A. We will associate four points with the singularity, as above. The convex segments of the two cells of Figure 2.12(b) are now $\widehat{PV_1}$, $\widehat{V_2Q}$, $\widehat{RW_1}$, and $\widehat{W_2S}$. If a traversal of the curve during a sort reaches P, then it will proceed to V_1 and then on from V_2 to Q. In particular, there is no danger of the traversal mistakenly proceeding from P to A to S.

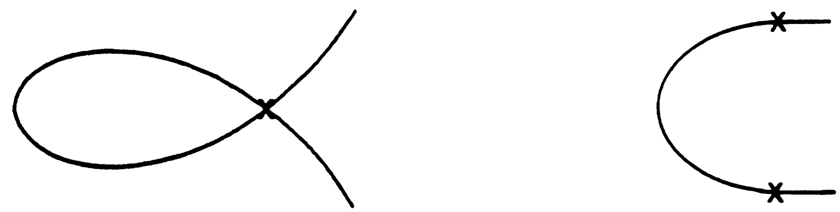
Notice that, after each singularity of the curve has been decomposed in this manner, every convex segment of the curve is bounded by simple points. Care must be taken with the sortpoints that lie on any of the segments about a singularity that are essentially sliced out, such as sortpoints that lie on $\widehat{V_1V_2}$ in the previous example. We shall discuss how to sort these points at the end of this section.

2.4.1 Blowing Up The Curve at a Singularity

For each branch that passes through a singularity, we wish to find a pair of points on it, one on either side of the singularity. We would like to do this by crawling a small distance along the arc in both directions from the singularity. However, there is no reliable way of crawling along a given arc as it passes through a singularity. Therefore, we must isolate each arc of the singularity so that we can crawl along it robustly. We accomplish this by blowing up the curve at the singularity by a series of quadratic transformations [9,35].

Suppose that a curve has been translated so that one of its singularities is at the origin. Let its new equation be $f(x,y) = 0$. Consider the **affine quadratic transformation** $x = x_1,\ y = x_1 y_1$ (of Cremona) [35] and the associated curve $f(x_1, x_1 y_1)$. The useful property of this transformation is that it maps the origin to the entire y_1-axis and maps the rest of the y-axis to infinity: $y_1 = \frac{y}{x}$ so $(0,b)$ maps to $(0, \frac{b}{0})$, which is a point at infinity unless $b = 0$. The quadratic transformation is one-to-one for all points (x,y) with $x \neq 0$. The line $y = mx$ through the origin is mapped to the horizontal line $y_1 = m$: $y = mx \rightarrow x_1 y_1 = m x_1 \rightarrow y_1 = m$. Thus, a quadratic transformation maps distinct tangent directions of the various branches of f at the singular origin to different points on the **exceptional line** $x_1 = 0$.[7] The intersections of the transformed branches with the exceptional line correspond to the transformed points of the singularity at the origin (Figure 2.13). If an intersection point on the exceptional line is singular, then the procedure is applied recursively (Figure 2.14). Hence,

[7]The quadratic transformation does not map the line $x = 0$ properly, so we must make sure that $x = 0$ is not a tangent direction to the curve at the origin. This is done by a nonsingular linear transformation $x = \alpha\hat{x} + \beta\hat{y}$ and $y = \delta\hat{x} + \gamma\hat{y}$, such that neither $\alpha\hat{x} + \beta\hat{y}$ nor $\delta\hat{x} + \gamma\hat{y}$ are tangents to the curve at the origin.



(a) (b)

Figure 2.13: (a) node and (b) its quadratic transformation

under quadratic transformations, the various branches of the curve in the neighbourhood of the singularity eventually get transformed to separate branches. Once a branch is isolated, it is simple to find two points of it on either side of the singularity, since there are no other branches present to cause confusion.

Lemma 2.1 **([4,35])** *A finite number of applications of the quadratic transformation reduces a singularity to a number of simple points.*

To summarize, for each singular point of the curve, we translate the singularity to the origin and apply a series of quadratic transformations until the singularity is transformed into a set of nonsingular points. Each branch of the transformed curve will intersect the exceptional line in a simple (nonsingular) point. For each of these branches, we compute two points on either side of the exceptional line $x_1 = 0$ (by crawling an ϵ-distance) and map this pair back onto the corresponding branch of the original curve, by

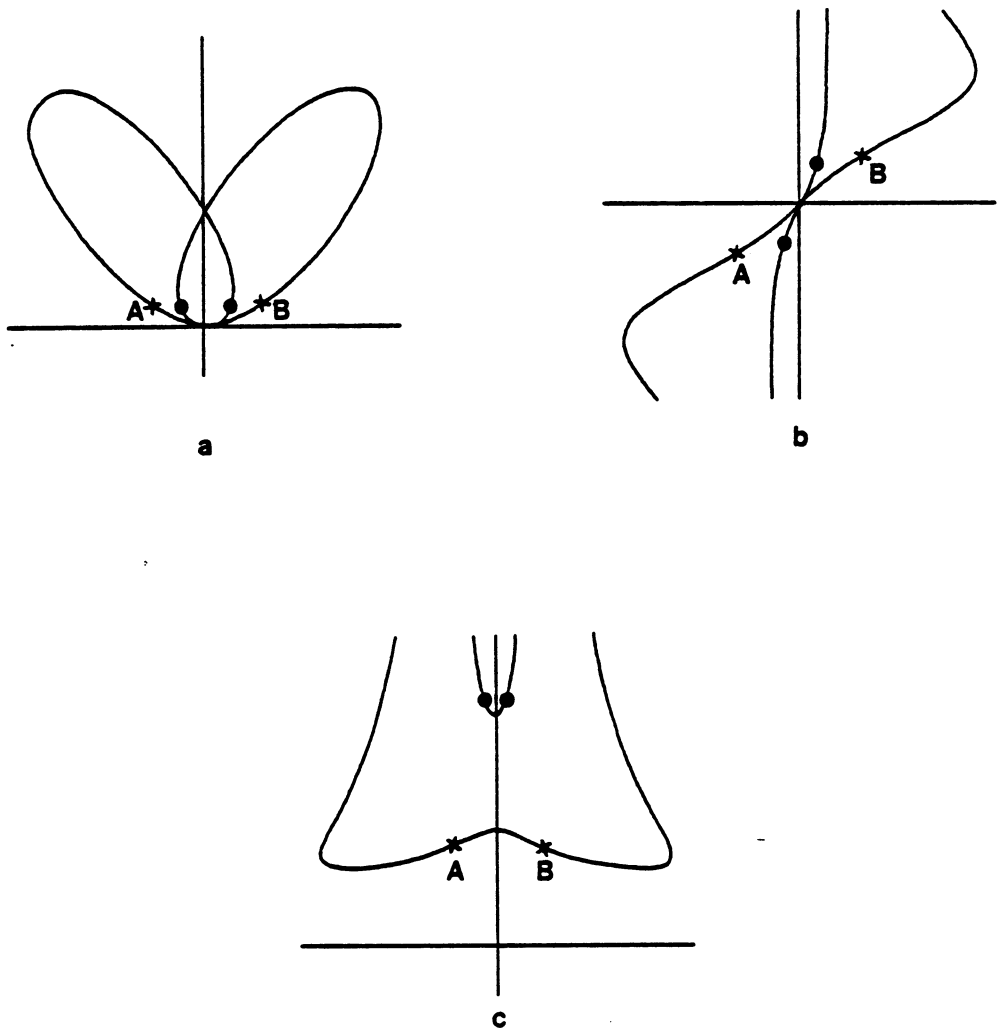


Figure 2.14: (a) the original singularity (b) after one quadratic transformation (c) after a second transformation: the original singularity has been successfully transformed into two simple points

applying inverse transformations. The pair of points on each branch clarify the branch connectivity at the singularity and allow a robust traversal of the curve by convex segments.

Definition 2.6 *The collection of points associated with a singularity are called* **pseudo curve points**. *(For example, V_1, V_2, W_1, and W_2 of Example 2.4.2 are pseudo curve points.) They replace the singularity curve point in the cell partition. That is, for the purposes of curve traversal, the singularity is no longer considered to be a curve point.*

As mentioned above, the sortpoints that lie close to a singularity (on one of the segments that is 'sliced out') must be treated as a special case. These points must be sorted by mapping them to the blown-up, desingularized curve and using the crawling method. This is not expensive because the sliced-out segment is very short and very few jumps are needed to crawl over it.

2.5 Of Flexes and Singularities

This section explains how to find the walls of a cell partition. We start with a discussion of projective space so that we can show how to find the singularities and flexes of a curve. We then examine how to find the tangents of a singularity and a flex, and how to distinguish a flox from a flex.

We have been working in **affine space**, the familiar n-dimensional Euclidean space in which points are represented as n-tuples, such as (0,0) for the origin of the plane. Projective space is an extension of affine space. Consider the equivalence relation $(x_1, \ldots, x_{n+1}) \approx (y_1, \ldots, y_{n+1})$ if and only if there exists $t \neq 0$ such that $x_i = t * y_i$ for all i. **N-dimensional projective space** (over the field K) is the space of equivalence classes of tuples of $K^{n+1} \setminus \{(0, 0, \ldots, 0)\}$ under this equivalence relation. The point $(x_1, \ldots, x_n)$ of affine space is identified with the point $(t * x_1, \ldots, t * x_n, t)$ $= (x_1, \ldots, x_n, 1)$ of projective space. That is, points in affine space are associated with points of the complement of $x_{n+1} = 0$ in projective space. The plane $x_{n+1} = 0$ of projective space is called the **plane at infinity**. It allows the formal treatment of points and components at infinity, such as the intersection of two curves at infinity. The point $(x_1, \ldots, x_n, 0)$ represents the point at infinity along the vector $(x_1, \ldots, x_n)$. Projective space is the extension of affine space by the plane at infinity.[8]

In projective space, a plane algebraic curve is the zero set of a **homo-**

[8]A more spatially oriented and evocative view of projective space is offered by Fulton [14]. The point $(x_1, \ldots, x_n)$ is identified with the line in K^{n+1} through $(0, 0, \ldots, 0)$ and $(x_1, \ldots, x_n, 1)$. Projective space is the collection of lines through $(0, 0, \ldots, 0)$ in K^{n+1}. Affine space is embedded in projective space as the plane $x_{n+1} = 1$. Two points of K^{n+1} on the same line are equivalent. The lines through $(0, 0, \ldots, 0)$ in the plane $x_{n+1} = 0$ correspond to the points at infinity.

geneous polynomial (i.e., a polynomial whose terms are all of the same degree) in three variables.[9] The curve $F(X, Y, Z) = 0$ in projective space corresponds to the curve $f(x, y) := F(x, y, 1) = 0$ in affine space. Conversely, the curve $f(x, y) = 0$ of order n in affine space corresponds to the curve $F(X, Y, Z) := Z^n f(\frac{X}{Z}, \frac{Y}{Z}) = 0$ in projective space, which is a homogenized version of $f(x, y) = 0$. In both cases, the curve $f(x, y) = 0$ is equivalent to the curve $F(X, Y, Z) = 0$ without its points at infinity.

Example 2.5.1 *The projective equivalent of the curve* $x^2 + 2x - 4y - 3 = 0$ *is* $x^2 + 2xz - 4yz - 3z^2 = 0$. *Consider the hyperbola* $x^2 - y^2 - 1 = 0$ *and one of its asymptotes* $x - y = 0$. *Their projective equivalents are* $x^2 - y^2 - z^2 = 0$ *and* $x - y = 0$, *respectively. Solving for their intersection yields* $x = y$, $z = 0$. *Therefore, since* $(x, x, 0) \approx (1, 1, 0)$, *the hyperbola intersects its asymptote at the point of infinity* $(1, 1, 0)$.

The following lemma gives the mathematical characterization of a singularity and a flex.

Lemma 2.2 ([35, pp. 51,71]) *Let* $F(x, y, z) = 0$ *be the representation of a plane curve* C *in projective space, and let* P *be a point of projective space.*

1. *P is a singularity of C if and only if* $F_x(P) = F_y(P) = F_z(P) = 0$

[9] The equation must be homogeneous, since solutions must be invariant under the above equivalence relation.

2. *P is a flex of C if and only if F(P) = 0, P is not a singularity, and* $det(F_{ij}(P)) = 0$,

where F_x *is the derivative of F with respect to x, etc.;* $F_{11} = F_{xx}$, *etc.; and* $(F_{ij}(P))$ *is a 3 by 3 matrix. (The curve* $det(F_{ij}(P)) = 0$ *is called the Hessian of the curve* $F = 0$.*)*

Therefore, the computation of the singularities of a curve of order n involves the solution of a system of three equations of degree $n - 1$. The computation of the flexes involves the solution of a system of two equations, one of degree n and the other of degree $3(n - 2)$. Resultants offer a possible method of solution (Appendix A). The solution of systems of equations is a well-studied problem, so we do not elaborate further.

The following lemma gives an indication of the worst-case complexity of a cell partition.

Lemma 2.3 ([35, pp. 65,120]) *A curve of order n has at most* $\frac{(n-1)(n-2)}{2}$ *singularities. A curve of order n that has no extraordinary*[10] *singularities of multiplicity greater than two has at most* $3n(n-2) - 6\delta - 8\kappa$ *flexes, where* δ *is the number of nodes of the curve (properly counted) and* κ *is the number of cusps.*

These maxima are not attained simultaneously. The maximum number

[10] See Appendix A.

of flexes occurs when there are no singularities, and the number of flexes decreases as the number of singularities increases.

It is simple to find the tangent of a flex.

Lemma 2.4 ([35, p. 55]) *If P is a nonsingular point of the plane curve $F(x,y,z) = 0$ (in projective space), then the equation of the tangent to F at P is*

$$F_x(P)x + F_y(P)y + F_z(P)z = 0$$

Definition 2.7 *The* **order form** *of a polynomial $f(x,y) = 0$ is the homogeneous polynomial consisting of the terms of lowest degree in f.*

The tangents of a singularity are found by translating the singularity to the origin and applying the following lemma.

Lemma 2.5 ([35, p. 54]) *If the order form of $f(x,y) = 0$ is of degree r, then the origin is a singularity of $f = 0$ of multiplicity r, and the components of the order form are the tangents to f at the origin.*

We wish to ignore all flexes of even order, so we require a method of computing the order of a flex.

Lemma 2.6 *Let P be a nonsingular point of a plane curve. Let $f(x,y) = 0$ be the equation of the curve after it has been translated and rotated so that P*

is the origin and P's tangent is the x-axis. Then the number of intersections of P's tangent with the curve at P is

$$min\ \{\ i \mid Ax^i \text{ is a term of } f(x,y),\ \text{for some } A \neq 0\}$$

Proof (This proof is a variation upon the discussion of [35, p. 53].) The intersections of $f(x,y)$ with the x-axis (whose parameterization is $x = t,\ y = 0$) are represented by the roots of $f(t,0) = 0$. The number of intersections of $f(x,y)$ with the x-axis at the origin is equal to the multiplicity of the root $t = 0$ in $f(t,0) = 0$. The Taylor expansion of $f(t,0)$ is

$$f(0,0) + f_x(0,0) * t + \frac{1}{2!}f_{xx}(0,0) * t^2 + \frac{1}{3!}f_{xxx}(0,0) * t^3 + \ldots$$

and the multiplicity of the root $t = 0$ is the lowest nonzero power of t in this expansion. That is, the multiplicity of intersection of $f(x,y)$ with the x-axis at the origin is

$$\min\ \{\ i \mid (\frac{df}{dx^i})(0,0) \neq 0\}$$

$$= \min\ \{\ i \mid Ax^i \text{ is a term of } f(x,y),\ \text{for some A} \neq 0\}$$

∎

2.6 Space Curves

The convex-segment method has been presented as a technique for sorting plane curves. Indeed, its reliance upon a cell partition of the curve's plane suggests that it can only be used to sort plane curves. However, we shall now show that space curves can also be sorted by the new method.

We solve a sorting problem for a space curve by mapping it to a related sorting problem for a plane curve. The plane curve is derived by projecting the space curve.

Definition 2.8 *Let B be a plane and let q be a point, $q \notin B$. The* **(central) projection of a point** *p (with respect to q and B) is the point of intersection of the line $\overleftrightarrow{pq}$ with B. The* **(central) projection of a space curve** *C (with respect to q and B) is the plane curve generated by the projections of the points of C. B is the* **plane of projection** *and q is the* **center of projection.**

Lemma 2.7 *Let $z = 0$ be the plane of projection and let (x_c, y_c, z_c) be the center of projection ($z_c \neq 0$). The projection map $P : \Re^3 \rightarrow \Re^2$ is*

$$P(x_{3d}, y_{3d}, z_{3d}) = (\frac{x_c z_{3d} - x_{3d} z_c}{z_{3d} - z_c}, \frac{y_c z_{3d} - y_{3d} z_c}{z_{3d} - z_c})$$

The inverse map $P^{-1} : \Re^2 \rightarrow \Re^3$ is

$$P^{-1}(x_{2d}, y_{2d}) = (\frac{x_{2d}(z_c - z_{3d}) + x_c z_{3d}}{z_c}, \frac{y_{2d}(z_c - z_{3d}) + y_c z_{3d}}{z_c}, z_{3d})$$

The inverse image of (x_{2d}, y_{2d}) must be expressed in terms of z_{3d}, since infinitely many points project to a given point on the plane (viz., all of the points on the line between $(x_{2d}, y_{2d}, 0)$ and (x_c, y_c, z_c)).

Proof The line through (x_c, y_c, z_c) and (x_{3d}, y_{3d}, z_{3d}) can be parameterized by $((x_{3d} - x_c)t + x_c, (y_{3d} - y_c)t + y_c, (z_{3d} - z_c)t + z_c)$. The projection of (x_{3d}, y_{3d}, z_{3d}) lies on this line and has a z-coordinate of 0. Therefore, it corresponds to the parameter value $\hat{t}$ where $(z_{3d} - z_c)\hat{t} + z_c = 0 \Rightarrow \hat{t} = \frac{-z_c}{z_{3d} - z_c}$. Therefore,

$$\text{proj } (x_{3d}, y_{3d}, z_{3d}) = ((x_{3d} - x_c)(\frac{-z_c}{z_{3d} - z_c}) + x_c,\ (y_{3d} - y_c)(\frac{-z_c}{z_{3d} - z_c}) + y_c,\ 0)$$

$$= (\frac{z_c(x_c - x_{3d}) + x_c(z_{3d} - z_c)}{z_{3d} - z_c}, \frac{z_c(y_c - y_{3d}) + y_c(z_{3d} - z_c)}{z_{3d} - z_c}, 0)$$

$$= (\frac{x_c z_{3d} - z_c x_{3d}}{z_{3d} - z_c}, \frac{y_c z_{3d} - z_c y_{3d}}{z_{3d} - z_c}, 0)$$

The inverse map can be derived by solving for x_{3d} and y_{3d} in the equations

$$x_{2d} = \frac{x_c z_{3d} - x_{3d} z_c}{z_{3d} - z_c},\ y_{2d} = \frac{y_c z_{3d} - y_{3d} z_c}{z_{3d} - z_c}$$

∎

Corollary 2.3 *Let $z = 0$ be the plane of projection and let (x_c, y_c, z_c) be the center of projection ($z_c \neq 0$). Let A be the plane $z = z_c$. The projection map $P : \Re^3 \rightarrow \Re^2$ is continuous on $\Re^3 \setminus A$. (That is, $lim_{\alpha \rightarrow 0}\ P(X + \alpha) = P(X)$ for $X \in \Re^3 \setminus A$.)*

Proof If f(x), g(x), and h(x) are polynomials, then the rational map $(\frac{f(x)}{h(x)}, \frac{g(x)}{h(x)})$ is continuous at all points where the denominator is not zero.

∎

Definition 2.9 *Order on the segment SEG is* **preserved** *by the projection map proj:*$\Re^3 \to \Re^2$ *if proj(SEG) is connected and, for all* $P_1, P_2, P_3 \in SEG$*, proj(*P_2*) lies in between proj(*P_1*) and proj(*P_3*) whenever* P_2 *lies in between* P_1 *and* P_3*.*

We want to choose a projection that preserves order on the sort segment of the space curve. We do this by ensuring that the projection map is continuous on the entire sort segment. In the rest of this section, F is a space curve defined by the intersection of the two (affine) surfaces $f_1(x, y, z) = 0$ and $f_2(x, y, z) = 0$, and $\widehat{PQ}$ is a finite segment of F (viz., the sort segment).

Lemma 2.8 *Let* $proj_C$ *be the projection map for the projection onto the plane* $z = 0$ *from the center of projection* C*. There exists a point* $C \in \Re^3$ *such that order on* $\widehat{PQ}$ *is preserved by the projection* $proj_C$*.*

Proof Let $\alpha = \max\{ z' \mid (x', y', z') \in \widehat{PQ}\}$. α exists because $\widehat{PQ}$ is a finite segment. Choose C = (x_C, y_C, z_C) so that $z_C > \alpha$. proj_C will be continuous on $\widehat{PQ}$, by Corollary 2.3. $\text{proj}_C(\widehat{PQ})$ is connected, since it is the continuous image of a connected segment. Let P_1, P_2, P_3 be points of $\widehat{PQ}$ such that P_2 lies in between P_1 and P_3. The continuity of proj_C ensures that $\text{proj}_C(P_2)$ lies in between $\text{proj}_C(P_1)$ and $\text{proj}_C(P_3)$. ∎

In solid modeling applications, $\widehat{PQ}$ will usually be the finite segment of a space curve that defines an edge of a solid model. The model will be bounded and, in particular, it will be bounded in the z direction. Therefore, the center of projection can be chosen above this bound to guarantee the continuity of the projection map on the sort segment, and thus the preservation of order on the projected sort segment. In the rest of this section, let $z = 0$ be the plane of projection and let (x_c, y_c, z_c) be the center of projection, chosen so that the projection map is continuous on $\widehat{PQ}$.

In order to apply the convex-segment method to the projection proj($\widehat{PQ}$) of the sort segment, we must know the implicit equation of the irreducible component of the projection that contains proj($\widehat{PQ}$). The following lemma shows how this equation can be determined from the equation $f_1{=}0 \cap f_2{=}0$ of the original space curve. It makes use of the resultant of a pair of polynomials, which is defined in Appendix A.

Lemma 2.9 *The projection of F is contained in the plane curve defined by the resultant $R(x,y)$ of $g_1(x,y,z) := f_1(\frac{x(z_c-z)+x_c z}{z_c}, \frac{y(z_c-z)+y_c z}{z_c}, z)$ and $g_2(x,y,z) := f_2(\frac{x(z_c-z)+x_c z}{z_c}, \frac{y(z_c-z)+y_c z}{z_c}, z)$ with respect to z. Moreover, the projection of $\widehat{PQ}$ is contained in a connected component of the curve $R(x,y) = 0$.*

Proof Let $\beta = (x_{2d}, y_{2d}, 0)$ be a point of the projection of F. By Lemma 2.7, the point of F that projects into β is $(\frac{x_{2d}(z_c-z_{3d})+x_c z_{3d}}{z_c}, \frac{y_{2d}(z_c-z_{3d})+y_c z_{3d}}{z_c}, z_{3d})$, for some z_{3d}. That is, there exists z_{3d} such that

$$f_1(\frac{x_{2d}(z_c - z_{3d}) + x_c z_{3d}}{z_c}, \frac{y_{2d}(z_c - z_{3d}) + y_c z_{3d}}{z_c}, z_{3d}) = 0$$

and

$$f_2(\frac{x_{2d}(z_c - z_{3d}) + x_c z_{3d}}{z_c}, \frac{y_{2d}(z_c - z_{3d}) + y_c z_{3d}}{z_c}, z_{3d}) = 0$$

Therefore, for every point (x_0, y_0) of the projection of F, there exists z_0 such that $g_1(x_0, y_0, z_0) = g_2(x_0, y_0, z_0) = 0$. Thus, by Lemma B.1, for every point (x_0, y_0) of the projection, $R(x_0, y_0) = 0$. Therefore, $R(x, y) = 0$ contains the projection of F. Since the projection map is continuous on $\widehat{PQ}$ and the continuous image of a connected set is connected, the projection of $\widehat{PQ}$ is connected. ∎

The problem of sorting points along a space curve has now been successfully reduced to a problem of sorting points along a plane curve. The equation of the plane curve is computed by the method of Lemma 2.9, and the sortpoints, start point, and end point are the projections of their counterparts on the space curve.

Let A be the plane that contains the center of projection and lies parallel to the plane of projection. By the choice of the center of projection (Lemma 2.8), all of the sort segment lies on the same side of A; and a sortpoint will lie on the same connected component as the sort segment if and only if it lies on this side of A. Therefore, since we want all of the projected sortpoints to lie on the same connected component of the projected plane curve (Section 1.2), we discard all sortpoints on the space curve that lie on the opposite side of A from the sort segment. These sortpoints do not lie on the sort segment, so they can be safely ignored.

If the resultant of Lemma 2.9 is factored into irreducible components (perhaps by the method of [8] or [36]), then it is simple to determine the component associated with the projection of the sort segment. Let proj(P) be the projection of a sortpoint of the space curve (one that has not been discarded). The desired component is the unique one that contains proj(P).

It is possible that a sortpoint of the space curve could project into a singularity. This must be avoided because the sorting of a set of points that includes singularities can be ambiguous (Section 1.2). Therefore, if a projected sortpoint proj(P) is discovered to be a singularity when the singularities and flexes of the projection are computed, then proj(P) is offset along the appropriate branch of the singularity by crawling a short distance along the space curve from the original sortpoint P to a point P_ϵ. That is, if proj(P) is a singularity, then we map P to proj(P_ϵ), which is not a singularity. (The crawl is stable because P cannot be a singularity. Care must be taken to make the crawl on the space curve short so that it does not crawl over another sortpoint, since this would disrupt the order of the sortpoints.)

Similarly, if two sortpoints S_1 and S_2 of the space curve project to the same point of the projection, then S_1 should be mapped to proj(S_1^ϵ) and S_2 to proj(S_2^ϵ), where S_i^ϵ is found by crawling a short distance from S_i. (Care must be taken that S_1^ϵ and S_2^ϵ do not still map to the same point.)

The computation of the resultant of Lemma 2.9 and its subsequent factorization (in order to find the irreducible component that contains the projection) are expensive operations. The expense of the factorization is the lesser problem, since the resultant is often already irreducible. We conclude that although the convex-segment method can indeed be used to sort points on a space curve, space curve sorting is considerably more expensive than plane curve sorting.

Since it can be difficult to find the parameterization of a space curve, the parameterization method may also decide to sort the projection of the space curve rather than the space curve itself. Recall that a rational parameterization of the space curve $S_1 \cap S_2$ is derived from a low degree, rational parameterization of S_1 or S_2. If there is no such parameterization (or no such parameterization is computable by a known algorithm), then the only recourse may be to look for a parameterization of the projection and sort it instead. Therefore, the expense of projecting the space curve to a plane curve may have to be absorbed by the parameterization method as well.

2.7 A Broad Comparison of the Methods

We have been introduced to three methods of sorting points along an algebraic curve. The crawling method sorts the points by making short jumps along the curve. The parameterization method observes that the sorting of points on a line is simple and tries to unwind the curve into a line by parameterizing it. The convex-segment method borrows from both of these methods.

Like the crawling method, the convex-segment method leaps from one point to another along the curve (viz., from a curve point to its partner). However, its jumps are large while the crawling method's jumps must be very small. Moreover, once the partner of each curve point of the cell partition has been computed (which can be done once and for all in a preprocessing step), each jump of the convex-segment method can be done very quickly; whereas, the crawling method must grope for some time (by applying Newton's method) to find the destination of each jump. In short, the convex-segment method makes large, bold jumps while the crawling method makes small, timid ones.

The convex-segment method is similar to the parameterization method because they both reduce the sorting problem to an easier one. However, rather than trying to reduce the entire problem (from sorting points on a curve to sorting real numbers), the convex-segment method divides the problem up into many smaller ones and reduces each one of these (from sorting points on a convex segment to sorting the angles those points make with a central point). We shall see that the many small reductions of the convex-segment method can be done more quickly than the single, large reduction of the parameterization method.

Chapter 3

Multisegment Cells

When a cell of a curve's cell partition contains more than one convex segment (Figure 3.1), two of the problems associated with curve sorting become

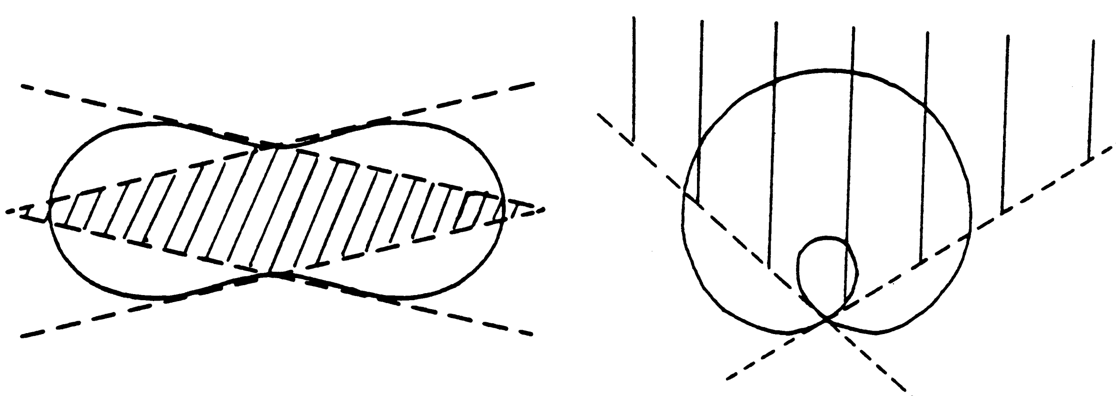


Figure 3.1: Some multisegment cells

nontrivial: determining the partners of a curve point and determining the convex segment that a sortpoint lies on. This chapter confronts these problems. Section 1 introduces the definitions and lemmas that are needed for the treatment of multisegment cells, while Sections 2 and 3 actually solve the problems.

3.1 Foundations

Throughout this section, C is a cell of the cell partition of a curve F, and P and Q are points of F.

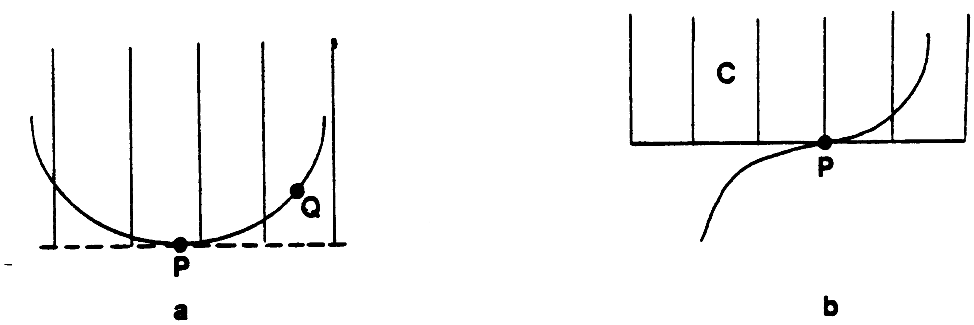


Figure 3.2: The inside of P's tangent

Definition 3.1 *If P is not a singularity or a flox, then* **the inside of P's tangent** *is the halfplane that contains all of the curve in the neighbourhood of P (Figure 3.2(a)). Otherwise, P's tangent is a wall of F's cell partition, and* **the inside of P's tangent with respect to the cell C** *is the halfplane that contains C (Figure 3.2(b)). The inside includes the tangent,* *while the* **strict inside** *does not.*

Let P be a flox that lies on the boundary of the cell C. Let P_ϵ *be a point of the curve inside cell C at distance* $\epsilon > 0$ *from P. (*P_ϵ *may be found by crawling into C from P.)* **The outside endpoint of P's cell segment with respect to C** *is the endpoint that lies outside of* P_ϵ*'s tangent, for* ϵ *small (Figure 3.3).*

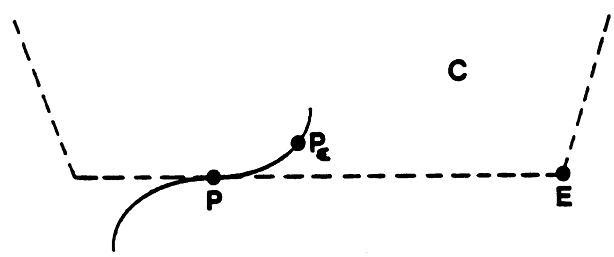


Figure 3.3: E is the outside endpoint of P's cell segment

If P is not a flox, then P **faces** *Q if Q lies on the inside of P's tangent (Figure 3.2(a)). Otherwise,* **P faces Q with respect to the cell C** *if (1) Q lies strictly inside P's tangent with respect to C, or (2) Q lies on P's tangent and Q lies on the opposite side of P from the outside endpoint of P's cell segment with respect to C (Figure 3.4).*

Notation 3.1 $\#\{S\}$ *is the number of elements in the set S.* $\overline{xy}$ *is the line segment between x and y, and it is assumed that* $\overline{xy}$ *does not include its*

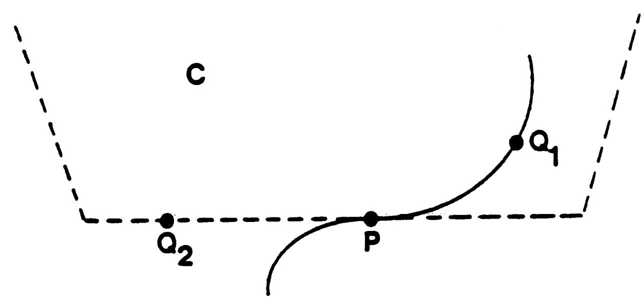


Figure 3.4: P faces both Q_1 and Q_2 with respect to C

endpoints x and y. Finally, we use 'w.r.t.' as an abbreviation for 'with respect to'.

The following lemma establishes three important properties of the line segment that joins two points of a convex segment. (Recall from Section 2.2 what it means for a line and a curve to cross.)

Lemma 3.1 *Consider the cell partition of a curve F, and a cell C of this partition. Let X and Y be two nonsingular points of a convex segment of the cell C. Then*

1. *the curve crosses $\overline{XY}$ at an even number of points (ignoring singularities)*

2. $\#\{P \in \overline{XY} \cap F : P \textit{ faces } X \textit{ w.r.t. } C\} =$
   $\#\{P \in \overline{XY} \cap F : P \textit{ faces } Y \textit{ w.r.t. } C\}$

3. for all $\alpha \in \overline{XY}$, $\#\{P \in \overline{X\alpha} \cap F : P$ *faces* X *w.r.t.* $C\} \leq$
$\#\{P \in \overline{X\alpha} \cap F : P$ *faces* Y *w.r.t.* $C\}$

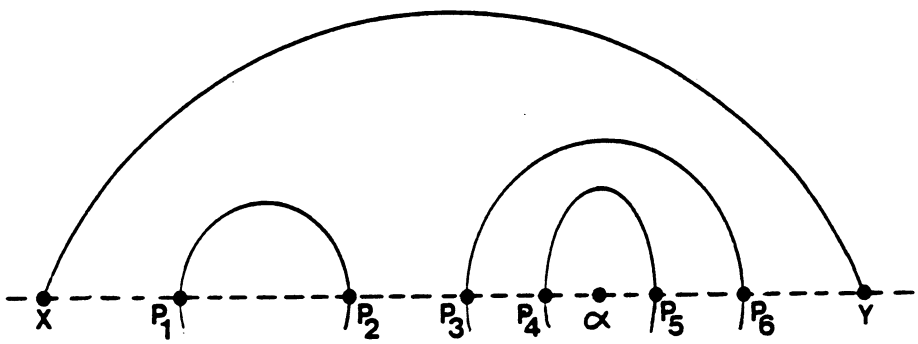


Figure 3.5: An example of Lemma 3.1

Example 3.1.1 *Figure 3.5 offers a hypothetical example for Lemma 3.1. The curve* F *crosses* $\overline{XY}$ *an even number of times.*

$$\{P \in \overline{XY} \cap F : \ P \textit{ faces } X\} = \{P_2, P_5, P_6\}$$

which is of the same size as

$$\{P \in \overline{XY} \cap F : \ P \textit{ faces } Y\} = \{P_1, P_3, P_4\}$$

Moreover,

$$\{P \in \overline{X\alpha} \cap F : \ P \textit{ faces } X\} = \{P_2\}$$

which is smaller than

$$\{P \in \overline{X\alpha} \cap F : \ P \textit{ faces } Y\} = \{P_1, P_3, P_4\}$$

Proof (of Lemma 3.1)

(1) Consider the closed region R_{XY} bounded by $\overline{XY}$ and $\widehat{XY}$ (Figure 3.6).

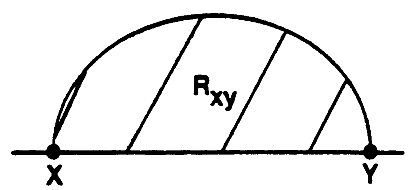


Figure 3.6: The region R_{XY}

R_{XY} lies in the cell C:

> $\widehat{XY}$ lies in the cell $\Rightarrow$ $\overline{XY}$ lies in the cell (since the cell is a convex polygon) $\Rightarrow$ the region bounded by $\widehat{XY}$ and $\overline{XY}$ lies in the cell (since the cell is a convex polygon)

X and Y are nonsingular, and $\widehat{XY}$ does not contain a singularity, since it lies in the interior of the cell. Therefore, the curve can only cross into R_{XY} through $\overline{XY}$. If the curve enters R_{XY}, then it must also leave R_{XY}, since an infinite segment cannot remain within a closed region (Lemma B.5) and an algebraic curve of finite length is closed (in particular, the curve cannot stop short in the middle of R_{XY}). We claim that the point of departure D must be distinct from the point of entry E, unless all of the tangents at $D = E$ are $\overleftrightarrow{XY}$, as in Figure 3.7. Otherwise, if $D = E$, then at least one of the tangents of the singularity D will cross into R_{XY} and form a wall of the cell partition which will split R_{XY} in two, contradicting the fact that all of R_{XY} lies in the same cell. Therefore, with the exception of the extraordinary singularities of Figure 3.7, the crossings of $\overline{XY}$ by the curve occur in pairs, which we shall call **couples**. This establishes condition (1) of the lemma.

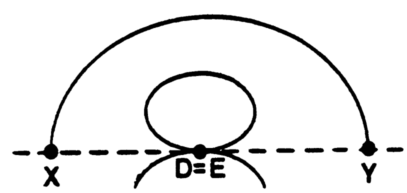


Figure 3.7: The only type of singularity that can lie on $\overline{XY}$

(2) Now consider condition (2) of the lemma. Note that the extraordinary singularities of Figure 3.7 can be ignored during the consideration of conditions (2) and (3), since they face both X and Y and contribute the same amount to the left-hand side and right-hand side of the expressions of conditions (2) and (3). Therefore, we can concentrate on the remaining crossings of $\overline{XY}$: the distinct 'couples'. Let $A, B \in \overline{XY}$ be a couple and assume, without loss of generality, that A lies closer to X than B does (Figure 3.8(a)). $\widehat{AB}$ is a convex segment since it lies within a cell of the cell partition. Therefore, A and B face each other (with respect to the cell C), by Lemma B.7. Since A faces B, A faces Y. Similarly, since B faces A, B

faces X. Therefore, one member of each couple faces X and the other faces Y. This yields the desired result:

$$\#\{P \in \overline{XY} \cap F : P \text{ faces X w.r.t. C}\} = \#\{P \in \overline{XY} \cap F : P \text{ faces Y w.r.t. C}\}$$

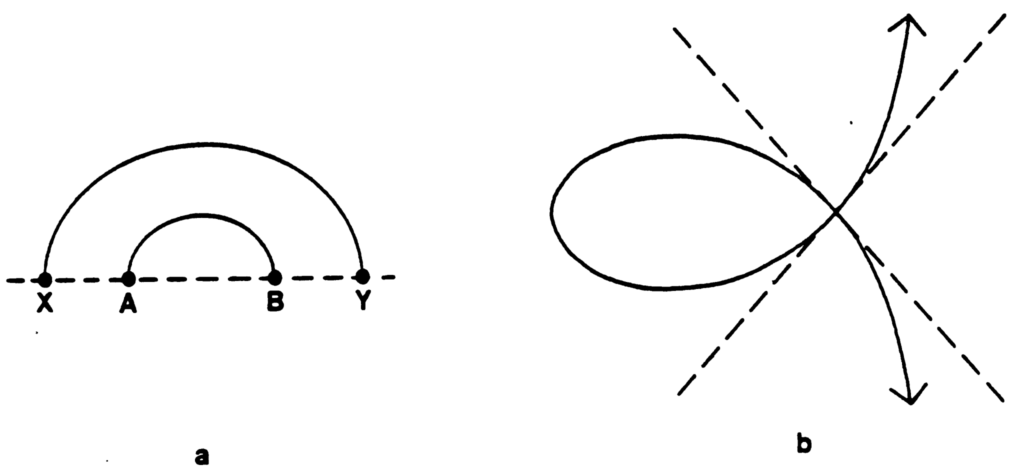


Figure 3.8: (a) a couple (b) open cells and open convex segments

(3) The point of a couple that faces Y (A) is closer to X than the point that faces X (B). Therefore, for all $\alpha \in \overline{XY}$,

$$\#\{P \in \overline{X\alpha} \cap F : P \text{ faces X}\} \leq \#\{P \in \overline{X\alpha} \cap F : P \text{ faces Y}\}$$

∎

3.2 Finding the Partners of a Curve Point

This section applies the theory of the previous section to the problem of determining the partners of a curve point. A solution of the partner problem is needed in order to find the next convex segment during a traversal of a curve by convex segments. (Recall that two curve points W_1 and W_2 of a cell partition are partners if $\widehat{W_1 W_2}$ is a convex segment.) Since we identify a convex segment by its two endpoints, if the partners of all of the curve points have been computed, then the two convex segments that leave a given curve point can be quickly identified.

Let F be a plane curve that has been split into convex segments by a cell partition. Consider a multisegment cell C of this cell partition and a curve point W_1 of this cell. Since singularities have been replaced by pseudo curve points (Section 2.4), W_1 is either a flox, an incidental curve point, or a pseudo curve point. Theorem 3.1 shows how to determine whether W_1 has a partner in C (i.e., whether W_1 is the endpoint of a closed convex segment in C) and, if W_1 does have a partner in C, how to find this partner. In preparation for this theorem, we must make some preliminary comments.

Definition 3.2 *A cell is* **closed** *(resp.,* **open***) if it is (resp., is not) a closed polygon (Figure 3.8(b)). An open cell is unbounded. A convex segment of the curve in cell C is* **closed** *if it is of finite length and* **open** *if it proceeds to infinity within C. Open segments have only one endpoint and* *must lie in open cells.*

The computation of W_1's partner involves the computation of intersections of lines with the boundary of cell C and a traversal of the cell boundary. Therefore, it is necessary for the cell to be closed. If C is an open cell, then temporary cell segments must be placed across its opening in order to artificially make it a closed cell (Figure 3.9). The added cell segments are called the **closing boundary** of C, and they must be chosen carefully. The resulting closed cell should be a convex polygon, it should be large enough to contain all of the closed convex segments of the original open cell, and it should have only one intersection with each open convex segment in the cell.

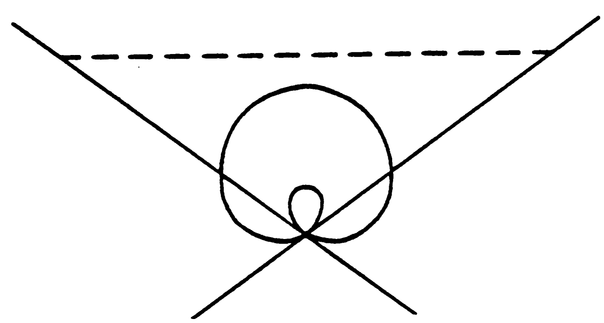


Figure 3.9: A closing boundary for an open cell

If C is an open cell that contains an open convex segment SEG, then we shall be interested in the intersection of this open segment with the closing boundary of C. If W_1 is the endpoint of SEG, then the intersection of SEG with the boundary will behave like a partner of W_1. Indeed, we shall identify that W_1 has no partner by noticing that the partner computed for W_1 lies on the closing rather than the original boundary of the cell. A point of intersection of the curve with the closing boundary shall be called a **closing curve point**.

There are now three families of curve points: (1) original curve points, which include floxes and incidental curve points; (2) pseudo curve points, which are the points that replace the singularities and guarantee a robust traversal of the curve; and (3) closing curve points, which are points on the closing boundary of open cells.

The computation of W_1's partner involves the sorting of curve points along the boundary of the cell C. However, a pseudo curve point does not lie on the boundary. Therefore, with each pseudo curve point W_i in C, we must associate a point W_i' on the cell boundary. If $W_i \neq W_1$, then W_i' is chosen to be the intersection of the ray $\vec{W_1 W_i}$ with the cell boundary (Figure 3.10(a)). (A link is maintained between W_i and W_i' so that it is simple to retrieve W_i from W_i'.) If W_1 is itself a pseudo curve point, then it has a special associated point. Let V be the singularity from which W_1 is derived, let T be the tangent to the branch of V that contains W_1, and let T_1 be the ray of W_1's tangent that intersects T (Figure 3.10(b)). $W_1{}'$ is the intersection of T_1 with the cell boundary. For notational consistency,

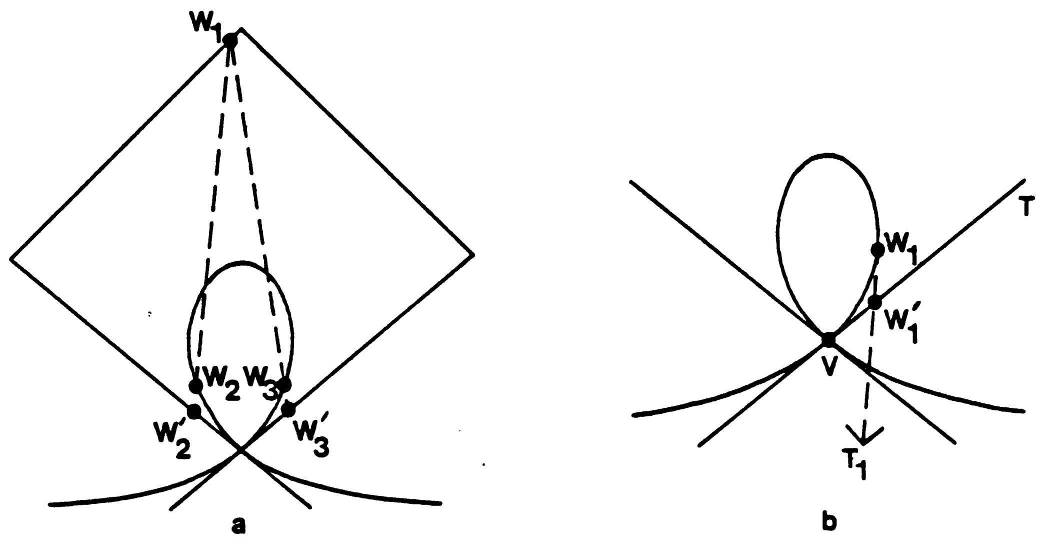


Figure 3.10: The boundary points W_i'

we let $W_i' = W_i$ if W_i is a curve point that already lies on the cell boundary (i.e., a flox, incidental, or closing curve point).

Finally, we partition the boundary of the cell into two regions. Let B_1 be the boundary of C from W_1' to X in one direction and B_2 the boundary from W_1' to X in the other direction, where X is defined as follows. If W_1 is a flox, let X be the outside endpoint of W_1's cell segment with respect to C (Definition 3.1 and Figure 3.11(a)). Otherwise, let $X \neq W_1'$ be the other intersection of W_1's tangent with the cell boundary (Figure 3.11(b-c)). We are now ready for the statement of the theorem.

Theorem 3.1 *Let* $S(W_1) = \{$*(original, pseudo, and closing) curve points* $W \neq W_1$ *of cell C of the cell partition of the curve F* $|$

1. *W lies on the strict inside of* W_1*'s tangent (with respect to C)*
2. $\#\{P \in \overline{W_1W} \cap F : P$ *faces* W_1 *(w.r.t. C)*$\} =$

   $\#\{P \in \overline{W_1W} \cap F : P$ *faces* W *(w.r.t. C)*$\}$
3. *for all* $\alpha \in \overline{W_1W}$,

   $\#\{P \in \overline{W_1\alpha} \cap F : P$ *faces* W_1 *(w.r.t. C)*$\} \leq$

   $\#\{P \in \overline{W_1\alpha} \cap F : P$ *faces* W *(w.r.t. C)*$\}$
4. *W faces* W_1 *(w.r.t. C)* $\}$

<u>Case 1</u>: *Suppose that* $S(W_1) \neq \emptyset$. *Let* $S'(W_1) = \{\ W' : W \in S(W_1)\ \}$. *Let* B_1 *and* B_2 *be the appropriate sections of the boundary of C, as defined*

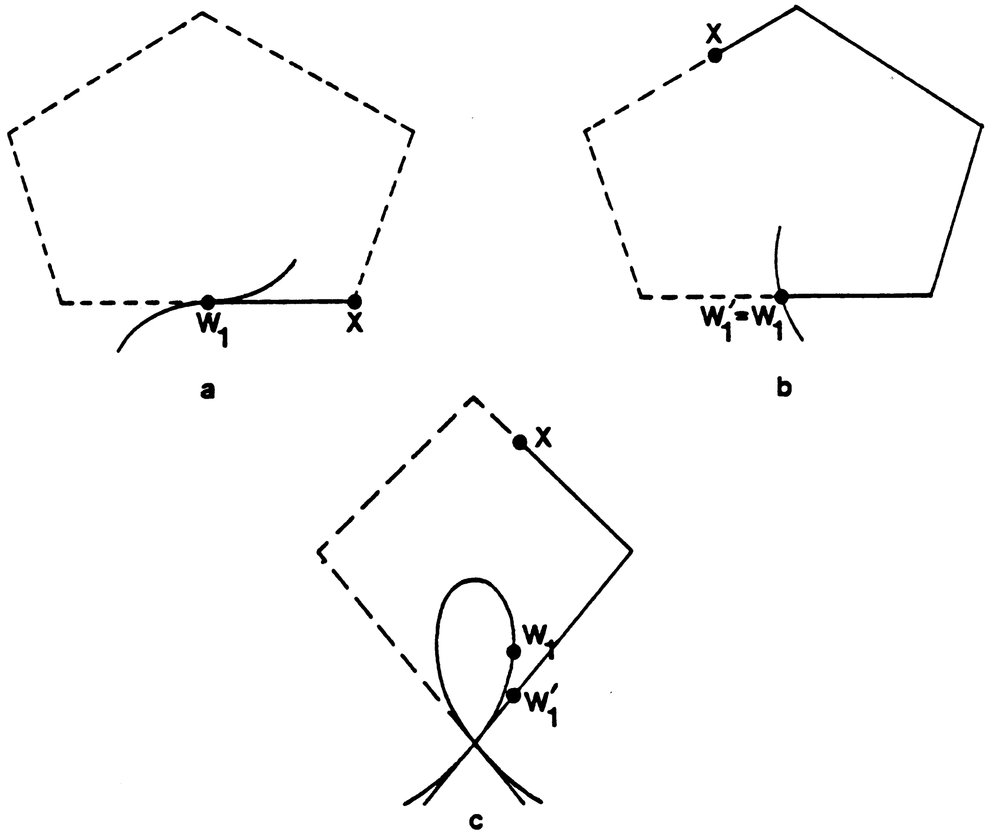


Figure 3.11: Partitioning the boundary of a cell

above. Either $S'(W_1) \subset B_1$ or $S'(W_1) \subset B_2$. (Assume without loss of generality that $S'(W_1) \subset B_1$.) Sort the points of $S'(W_1)$ along B_1 from W_1' to X. That is, sort the points of $S'(W_1)$ into $S_1', S_2', \ldots, S_p'$, where S_i' is encountered before S_{i+1}' in a traversal of the cell boundary from W_1' to X along B_1. If S_p (the curve point associated with S_p') is a closing curve point, then W_1 has no partner in C. Otherwise, S_p is W_1's partner in C.

<u>Case 2</u>: *Suppose that $S(W_1) = \emptyset$. Then W_1 is not a pseudo curve point*

and W_1's partner lies on W_1's cell segment. Let $T(W_1) = \{$ original curve points[1] *W of C |*

1. *W lies on W_1's cell segment*

2. *W_1 faces W and W faces W_1*

3. *$\#\{P \in \overline{W_1W} \cap F$: P faces W_1 (w.r.t. C)$\}$ =*
   $\#\{P \in \overline{W_1W} \cap F$: P faces W (w.r.t. C)$\}$ $\}$

W_1's partner is the element of $T(W_1)$ that is closest to W_1.

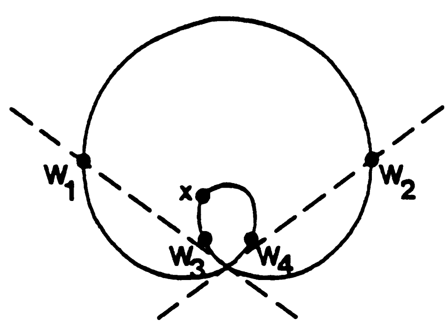


Figure 3.12: Cell partition of a limacon

Example 3.2.1 *Consider the cell partition of a limacon (Figure 3.12) and the multisegment cell containing the convex segments $\widehat{W_1W_2}$ and $\widehat{W_3W_4}$. Suppose that we wish to find the partner of W_1. W_3 violates condition (4)*

[1]That is, floxes and incidental curve points.

of S(W_1) and W_4 violates condition (2), so $S(W_1) = \{W_2\}$ and it is clear that W_2 is W_1's partner.

Consider the multisegment cell of Figure 3.13 and the computation of W_1's partner, where W_1 is the endpoint of an open convex segment. $S(W_1) = \{W_2, W_3, W_4\}$ and $S'(W_1) = \{W_2, W_3, W_4'\}$. The sorted order of $S'(W_1)$ along the boundary from $W_1{}'$ $(= W_1)$ to X (the intersection of W_1's tangent with the boundary) is W_3, W_4', W_2. The last element is W_2, which is a closing curve point. Therefore, W_1 has no partner.

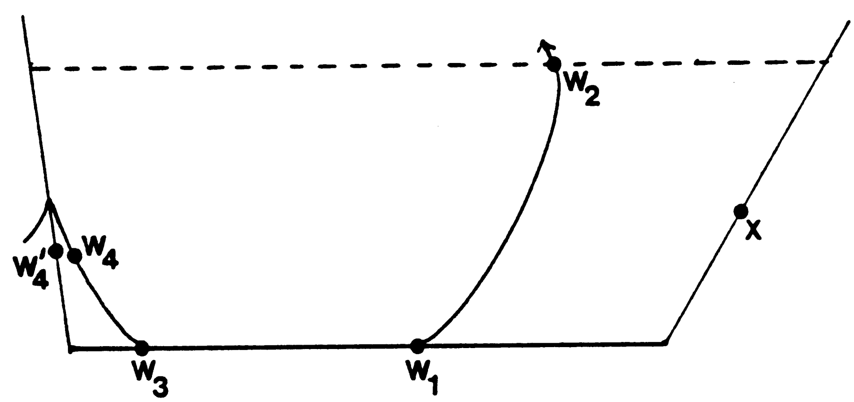


Figure 3.13: Computing the partner of the endpoint of an open convex segment

Finally, consider the computation of the partner of W_1 in Figure 3.14, where $S(W_1) = \emptyset$. V_1, V_2 and V_4 are ruled out by condition (2) of T(W_1), while V_3 and V_6 are ruled out by condition (3). Therefore, $T(W_1)$ =

$\{V_5, W_2\}$. W_2 is the closest element of $T(W_1)$ to W_1, so it is W_1's partner.

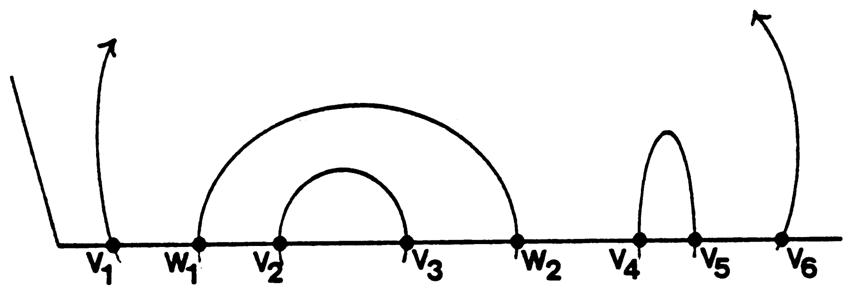


Figure 3.14: Partner computation when $S(W_1) = \emptyset$

Proof (of Theorem 3.1)

If W_1 is the endpoint of an open convex segment SEG, then let W_2 be the intersection of SEG with the closing boundary of the cell. Otherwise, let W_2 be W_1's partner.

Case 1: Suppose that $S(W_1) \neq \emptyset$. Let $\widehat{W_1W_2}$ be the boundary of the cell from W_1' to W_2', such that $X \notin \widehat{W_1W_2}$ (Figure 3.15(a)). $\widehat{W_1W_2}$ is a subset of either B_1 or B_2. We will show that $S'(W_1) \subset \widehat{W_1W_2}$. Let $s \in S(W_1)$.

Claim: $\vec{W_1s}$ does not cross $\widehat{W_1W_2} \setminus \{W_2\}$.

Proof of claim: Suppose, for the sake of contradiction, that $\vec{W_1s}$ crosses $\widehat{W_1W_2}$ at $y \neq W_2$. This is impossible if $s = W_2$ (since $\widehat{W_1W_2}$ is convex) so assume that $s \neq W_2$.

Subcase 1: Suppose that $s \in \overline{W_1y}$. By the argument of the proof of

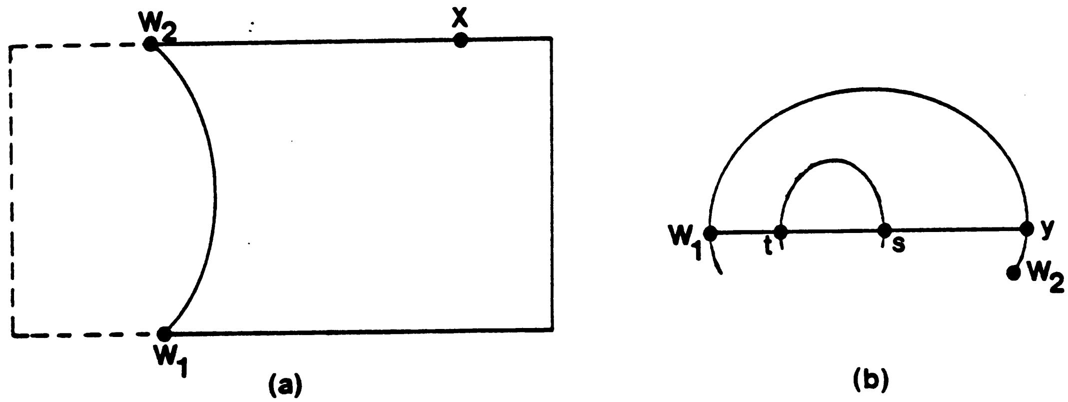


Figure 3.15: (a) $\widehat{W_1W_2}$ is dotted (b) s and t on $\overline{W_1y}$

Lemma 3.1, the points of intersection of the curve F with $\overline{W_1y}$ pair up. Let t be the partner of s (Figure 3.15(b)). The segment $\widehat{st}$ is convex, so s and t face each other (Lemma B.7). s faces W_1 (condition (4) of S(W_1)) and t, so $t \in \overline{W_1s}$. Since $s \in S(W_1)$,

$$\#\{P \in \overline{W_1s} \cap F: \text{ P faces } W_1\} = \#\{P \in \overline{W_1s} \cap F: \text{ P faces s}\}$$

Since $\overline{W_1s}$ does not include its endpoints, $\overline{W_1s} = \overline{W_1t} \cup \overline{ts} \cup \{t\}$. Also, t faces s. Therefore, the above equation becomes

$$\#\{P \in \overline{W_1t} \cap F: \text{ P faces } W_1\} + \#\{P \in \overline{ts} \cap F: \text{ P faces } W_1\} + 0 =$$

$$\#\{P \in \overline{W_1t} \cap F: \text{ P faces s}\} + \#\{P \in \overline{ts} \cap F: \text{ P faces s}\} + 1$$

Moreover, by Lemma 3.1 ($\widehat{st}$ is convex),

$$\#\{P \in \overline{ts} \cap F: \text{ P faces s}\} =$$

$$\#\{P \in \overline{ts} \cap F: \text{ P faces t}\} =$$

$$\#\{P \in \overline{ts} \cap F: \text{ P faces } W_1\}$$

Upon the cancellation of terms in the above equation, we conclude that

$$\#\{P \in \overline{W_1 t} \cap F: \text{ P faces } W_1\} >$$

$$\#\{P \in \overline{W_1 t} \cap F: \text{ P faces s}\} =$$

$$\#\{P \in \overline{W_1 t} \cap F: \text{ P faces y}\}$$

This contradicts condition (3) of Lemma 3.1 ($SEG = \widehat{W_1 W_2}$, $X = W_1$, $Y = y$).

Subcase 2: Suppose that $y \in \overline{W_1 s}$. By Lemma 3.1,

$$\#\{P \in \overline{W_1 y} \cap F: \text{ P faces } W_1\} = \#\{P \in \overline{W_1 y} \cap F: \text{ P faces y}\}$$

But y faces W_1, since W_1 and y are on the same convex segment. Therefore, there exists $\alpha \in \overline{W_1 s}$ such that

$$\#\{P \in \overline{W_1 \alpha} \cap F: \text{ P faces } W_1\} > \#\{P \in \overline{W_1 \alpha} \cap F: \text{ P faces s}\}$$

This is a contradiction of $s \in S(W_1)$ (condition (3)).

QED of Claim

We conclude that $\overrightarrow{W_1 s}$ does not cross $\overset{\frown}{W_1 W_2}\backslash\{W_2\}$. In particular, $\overline{W_1 s'}$ does not cross $\overset{\frown}{W_1 W_2} \setminus \{W_2\}$. Therefore, s' must either lie on $\overset{\frown}{W_1 W_2}$, outside of W_1's tangent, or on $\widehat{W_1 W_2}$. However, since s lies on the strict inside of W_1's tangent (by condition (1) of $S(W_1)$), so does s'. Moreover, the only curve points on $\overset{\frown}{W_1 W_2}$ are W_1 and W_2,[2] both of which lie on $\widehat{W_1 W_2}$. Therefore, s' must lie on $\widehat{W_1 W_2}$, and we have successfully shown that $S'(W_1) \subset \widehat{W_1 W_2}$.

We now show that $W_2 \in S(W_1)$.

(1) Suppose, for the sake of contradiction, that W_2 lies on W_1's tangent. By Lemma B.8, W_2 must lie on W_1's wall. Thus, $\widehat{W_1 W_2} = \overline{W_1 W_2}$, a subsegment of W_1's wall. Again by Lemma B.8, W_1 must be a flox whose tangent is a wall of the cell partition, so $\overline{W_1 W_2}$ is a subsegment of W_1's tangent. By condition (1) of $S(W_1)$, $S(W_1) \cap \overline{W_1 W_2} = \emptyset$. Therefore, $S'(W_1) \cap \overline{W_1 W_2} = \emptyset$. But $S'(W_1) \subset \widehat{W_1 W_2} = \overline{W_1 W_2}$. Thus, $S'(W_1) = \emptyset$, which is a contradiction. We conclude that W_2 does not lie on W_1's tangent. W_2 certainly lies on the inside of W_1's tangent, since $\overset{\frown}{W_1 W_2}$ is a convex segment.

(2-3) Lemma 3.1 (SEG = $\overset{\frown}{W_1 W_2}$, X = W_1, Y = W_2)

(4) Lemma B.7

[2]Recall the conditions that were placed on the closing boundary.

We are now prepared to show that $W_2 = S_p$. Since X is not on $\widehat{W_1W_2}$ (by definition), all of $\widehat{W_1W_2}$ is met in traversing the cell boundary from W_1' to X through $\widehat{W_1W_2}$. Therefore, since W_2' is an endpoint of $\widehat{W_1W_2}$ and all of $S'(W_1)$ is contained in $\widehat{W_1W_2}$, W_2' is the last element of $S'(W_1)$ that is met in traversing from W_1' to X through $\widehat{W_1W_2}$. In other words, $W_2' = S_p'$, and $W_2 = S_p$.

We must show that $S_i' \neq S_j'$ whenever $i \neq j$, so that there is no ambiguity in choosing the last member of $S'(W_1)$. Suppose that $i \neq j$ but $S_i' = S_j'$. Then $S_i, S_j \in \overline{W_1S_i'}$ and we can assume without loss of generality that $S_i \in \overline{W_1S_j}$. $S_i \in S(W_1)$ implies that $\#\{P \in \overline{W_1S_i} \cap F : P \text{ faces } W_1\}$ $= \#\{P \in \overline{W_1S_i} \cap F : P \text{ faces } S_i\} = \#\{P \in \overline{W_1S_i} \cap F : P \text{ faces } S_j\}$. Moreover, $S_i \in S(W_1)$ implies that S_i faces W_1. Therefore, there exists $\alpha \in \overline{W_1S_j}$ such that $\#\{P \in \overline{W_1\alpha} \cap F : P \text{ faces } W_1\} > \#\{P \in \overline{W_1\alpha} \cap F : P \text{ faces } S_j\}$, which is a contradiction of $S_j \in S(W_1)$. Therefore, $S_i' \neq S_j'$ if $i \neq j$, and the sort of $S'(W_1)$ is well-defined.

Case 2: Suppose that $S(W_1) = \emptyset$. We claim that if W_1 is a pseudo curve point, then W_2 is an element of S(W_1), which contradicts $S(W_1) = \emptyset$:

(1) of S(W_1) Since W_1 is a pseudo curve point, W_2 does not lie on W_1's tangent (Lemma B.8). W_2 lies inside W_1's tangent because $\widehat{W_1W_2}$ is convex.

(2-3) of S(W_1) Lemma 3.1 (X= W_1, Y= W_2)

(4) of $\mathbf{S}(W_1)$ Lemma B.7

Therefore, W_1 is not a pseudo curve point and it is well-defined to speak of W_1's cell segment. We show that $W_2 \in T(W_1)$:

(1) of $\mathbf{T}(W_1)$ Suppose that W_2 lies strictly inside W_1's wall (w.r.t. C). Then $W_2 \in S(W_1)$, a contradiction of $S(W_1) = \emptyset$:

(1) of $\mathbf{S}(W_1)$ The segment $\widehat{W_1 W_2}$ is convex, so W_2 lies on the inside of W_1's tangent. Since W_2 lies strictly inside W_1's wall, it cannot lie on W_1's tangent (Lemma B.8).

(2-3) of $\mathbf{S}(W_1)$ Lemma 3.1

(4) of $\mathbf{S}(W_1)$ Lemma B.7

Therefore, W_2 lies on W_1's wall.

(2) of $\mathbf{T}(W_1)$ Lemma B.7

(3) of $\mathbf{T}(W_1)$ Lemma 3.1

Therefore, $W_2 \in T(W_1)$.

Suppose that W_2 is not the closest member of $T(W_1)$ to W_1. Let $U \neq W_2$ be the closest. Since W_1 faces U, U must lie on $\overline{W_1 W_2}$. By the proof used in Lemma 3.1, the nonsingular points of intersection of the curve with $\overline{W_1 W_2}$ must pair up into couples, since $\widehat{W_1 W_2}$ is a nonsingular, convex segment. In particular, the original curve points on $\overline{W_1 U} \subset \overline{W_1 W_2}$ that face W_1 must pair with the equal number of original curve points on $\overline{W_1U}$ that face U. But U must also pair with a curve point on $\overline{W_1U}$ that faces U, and there are no such curve points remaining without a partner. This contradiction leads us to conclude that W_1's partner W_2 must be the closest element of T(W_1) to W_1. ∎

Since the pairing of the curve points of a curve does not depend upon the sortpoints, it can be done in a preprocessing phase. In particular, the creation of the cell partition and the computation of partners can be done at any time between the definition of the curve and the sorting of points along the curve.

3.3 Finding the Convex Segment that a Sortpoint Lies On

In this section, we present a solution to the second of our problems: determining the convex segment that a sortpoint lies on. This is a key step in the sorting of a set of points by the convex-segment method, since it offers a method of determining which points lie on a given convex segment during a traversal of the curve.

As in the previous section, let F be a plane curve that has been split into convex segments by a cell partition. Consider a multisegment cell C of this cell partition and a sortpoint x of the curve in the interior of C. (If x lies on the boundary of the cell, then Theorem 3.1 can be used to determine its partner and thus its convex segment.) The following theorem shows how to determine the convex segment of C that contains x. Since a convex segment is identified by its endpoints, determining the convex segment that x lies on is equivalent to finding the curve points that bound this convex segment. Therefore, Theorem 3.2 is very similar to Theorem 3.1, since they both involve finding the endpoints of a given point's convex segment.

As with Theorem 3.1, an open cell must be artificially closed, and a curve point W_i must have an associated boundary point W_i''. We choose W_i'' to be the intersection of $x\vec{W_i}$ with the cell boundary. We also need to partition the cell boundary into two regions again. Let B_1 be the boundary of the cell from x_1 to x_2 in one direction, and let B_2 be the boundary in the other direction, where x_1 and x_2 are the two points of intersection of x's tangent with the cell boundary.

Theorem 3.2 *Let* $S(x) = \{$*(original, pseudo, closing) curve points* W *of* $C \mid$

1. *W lies on the strict inside of x's tangent*
2. $\#\{P \in \overline{xW} \cap F : P \text{ faces } x\} = \#\{P \in \overline{xW} \cap F : P \text{ faces } W\}$
3. $\forall\ \alpha \in \overline{xW}$,

$$\#\{P \in \overline{x\alpha} \cap F : P \text{ faces } x\} \leq \#\{P \in \overline{x\alpha} \cap F : P \text{ faces } W\}$$

4. W faces x }

Let $S''(x) = \{ W'' : W \in S(x) \}$. *Either* $S''(x) \subset B_1$ *or* $S''(x) \subset B_2$. *(Assume without loss of generality that* $S''(x) \subset B_1$.*) Sort the points of* $S''(x)$ *into* $S''_1, S''_2, \ldots, S''_p$, *where* S''_i *is encountered before* S''_{i+1} *in a traversal of the cell boundary from* x_1 *to* x_2 *along* B_1. *Then either (i)* S_1 *and* S_p *are partners and x lies on the convex segment* $\widehat{S_1 S_p}$, *or (ii) x lies on an open convex segment SEG, one of* S_1, S_p *is a closing curve point, and the other is the endpoint of SEG.*

Example 3.3.1 *Consider the cell partition of the limacon (Figure 3.12) and the multisegment cell containing the convex segments* $\widehat{W_1 W_2}$ *and* $\widehat{W_3 W_4}$. *Suppose that we wish to know the convex segment that x lies on. We compute* $S(x)$. W_1 *does not satisfy condition (1), and* W_2 *does not satisfy condition (3). Thus,* $S(x) = \{W_3, W_4\}$ *and x must lie on* $\widehat{W_3 W_4}$.

Consider the cell partition of the Cassinian oval (Figure 2.10). Suppose that we wish to know the convex segment that P_1 *lies on. Since* $S(P_1) = \{W_1, W_2, W_3, W_4\}$, *it does not resolve the question. Let* x_1 *and* x_2 *be the two points of intersection of* P_1*'s tangent with the cell walls. The sort of* $S''(P_1)$ *from* x_1 *to* x_2 *is* W_1, W_3, W_4, W_2, *so* P_1 *must lie on* $\widehat{W_1 W_2}$.

Proof (of Theorem 3.2)

The proof is entirely analogous to the proof of Theorem 3.1. Let SEG be the convex segment that contains x. Let W_1 and W_2 be the endpoints of SEG or, if SEG is an open convex segment, let W_1 be its one endpoint and let W_2 be the intersection of SEG with the closing boundary of the cell. We first show that $W_1, W_2 \in S(x)$:

(1) Since $x \in \widehat{W_1 W_2}$, $\widehat{W_1 W_2}$ is convex, and $x \neq W_1, W_2$, both W_1 and W_2 must lie on the strict inside of x's tangent.

(2-3) Lemma 3.1 (SEG = $\widehat{W_1 W_2}$, X= x, Y= W_1 or W_2)

(4) Lemma B.7

Let $\widehat{W_1 W_2}$ be the boundary of the cell from W_1'' to W_2'', such that $x_1, x_2 \notin \widehat{W_1 W_2}$. $\widehat{W_1 W_2}$ is a subset of either B_1 or B_2. By the argument used in Theorem 3.1, $S''(x) \subset \widehat{W_1 W_2}$. $x_1, x_2 \notin \widehat{W_1 W_2}$ (by definition), so W_1'' and W_2'' are the first and last points of $S''(x) \subset \widehat{W_1 W_2}$ that are met in a traversal of the cell boundary from x_1 to x_2 through $\widehat{W_1 W_2}$. Therefore, $\{W_1'', W_2''\} = \{S_1'', S_p''\}$, and $\{W_1, W_2\} = \{S_1, S_p\}$. ∎

It can be expensive to create the set S(x). In particular, conditions (2) and (3) require the points of intersection of the curve with a line segment, which involves the solution of an equation of degree n, where n is the order of the curve. This is an expensive operation that we would like to avoid. Fortunately, it is usually possible to do so.[3]

[3]The expensive conditions cannot be avoided in Theorem 3.1. However, partner computation is a one-time preprocessing step. Moreover, conditions (1) and (4) will often rule out all of the curve points except the partner.

Before the more expensive conditions (2) and (3) of S(x) are tested, we would like to eliminate as many curve points as possible from contention. Therefore, we would like to find a collection of inexpensive conditions that must be satisfied by the endpoints of a sortpoint's convex segment. The inexpensive conditions that we choose are motivated by conditions (1) and (4) of S(x) and the following observations. First, as soon as the curve point W is eliminated, W's partner can also be eliminated, since the endpoints of a sortpoint's convex segment are partners. Second, all of the curve segment between a curve point W_1 and its partner W_2 lies on one side of $\overline{W_1W_2}$, since $\widehat{W_1W_2}$ is a convex segment. Thus, if the sortpoint does not lie on the appropriate side of $\overline{W_1W_2}$ (viz., the inside of the chord $\overline{W_1W_2}$, a term that is defined at the end of Appendix A), then both W_1 and W_2 can be eliminated.

We can now present a more efficient algorithm for finding the endpoints of a sortpoint x's convex segment. Let R(x) = {curve points W of cell C |

1. W and its partner in C lie on the strict inside of x's tangent
2. x lies on the strict inside of W's tangent and the strict inside of W's partner's tangent
3. x lies on the inside of the chord of W's convex segment in C}

R(x) contains the desired endpoints of x's convex segment. Therefore, we are finished if $R(x) = \{W_1, W_2\}$ and W_1 and W_2 are partners, or if

$R(x) = \{W_1\}$. However, if R(x) contains more than two curve points or if $R(x) = \{W_1, W_2\}$ and both W_1 and W_2 are endpoints of open convex segments, then we must revert to an application of Theorem 3.2. Since we already know R(x), it should be used to compute S(x) more efficiently: S(x) = $\{ W \in R(x) \mid$

1. $\#\{P \in \overline{xW} \cap F : P \text{ faces } x\} = \#\{P \in \overline{xW} \cap F : P \text{ faces W}\}$

2. for all $\alpha \in \overline{xW}$,

$$\#\{P \in \overline{x\alpha} \cap F : P \text{ faces } x\} \leq \#\{P \in \overline{x\alpha} \cap F : P \text{ faces W}\}\}$$

The problems associated with multisegment cells have now been solved. This completes our discussion of the theory of the convex-segment method of sorting points along an algebraic curve. In the next chapter, we turn to an analysis of its behaviour.

Chapter 4

A Comparison of the Sorting Methods

This chapter examines the efficiency of the convex-segment method of sorting. Section 1 analyzes its theoretical complexity, while Section 2 presents some empirical results for the sorting methods. Section 3 discusses the advantages of the convex-segment method.

4.1 Complexity Analysis

This section presents an analysis of the worst case complexity of the convex-segment method. We include this analysis for the insight that it offers into the algorithm. However, we must emphasize that it is often unwise to compare geometric modeling algorithms by their worst case performance, because worst cases can be misleadingly pessimistic and a geometric modeler is concerned about the treatment of cases that arise in practice rather than the behaviour of the algorithm on a worst case that occurs very rarely. For example, in the worst case, Theorem 3.2 will have to be applied in order to determine the convex segment that a given sortpoint x lies on, which could involve the solution of several equations of degree n. Yet, the worst case arises only in those rare cases when the sortpoint x lies in a multi-segment cell and requires the expensive conditions of $S(x)$ to determine its convex segment. (The use of $R(x)$, as described in Section 3.3, makes this worst case even more unlikely.) In all other cases, x's convex segment can be found in $O(1)$ time. An expected case analysis would be preferable, however it is difficult to formalize the notion of an expected case for sorting with the convex-segment method.

The complexity of sorting m points along a rational curve with the parameterization method is $O(m\alpha[p])$, where $\alpha[d]$ is the time required to find the real roots of a polynomial equation of degree d and p is the degree of the curve's rational parameterization. The complexity of finding a parameterization depends upon the algorithm used. However, it should be of the same order of complexity as finding the singularities and flexes of the curve, since singularities are used in the algorithm for parameterizing rational curves.

The complexity of sorting m points along a sort segment of a curve with the crawling method is $O(\frac{NL}{\epsilon})$, where N is the time required to apply Newton's method (which will vary from application to application), L is the length of the sort segment, and ϵ is the size of each jump.

Theorem 4.1 *m points on a plane algebraic curve of order n can be sorted by the convex-segment method in $O(mn^3\alpha[n])$ worst case time after $O(\alpha[n^2]+n^6\alpha[n]+n^2\alpha[2^{MAX}n]+n^4 2^{2*MAX})$ preprocessing, where $\alpha[n]$ is the time required to find the real roots[1] of a univariate polynomial equation of degree n and* MAX *is the maximum number of quadratic transformations that are necessary to decompose any singularity of the curve into simple points.*[2]

Proof The singularities of a curve are found by solving the simultaneous system of equations $\{f_x = 0, f_y = 0, f_z = 0\}$ (Lemma 2.2), which can be done effectively by using resultants. Let X be the real roots of the resultant of f_x and f_y with respect to y, which is a univariate[3] polynomial in x of degree $O(n^2)$. Similarly, let Y be the real roots of the resultant of f_x and f_y with respect to x. Then the singularities of the curve in the real, affine plane are $\{\ (x,y)\ :\ x \in X,\ y \in Y \text{ and } f_z(x,y) = 0\ \}$. A resultant of

[1]The procedure of Jenkins and Traub [21] for computing real roots is a good choice.

[2]*MAX* will usually be 1 or 2. For example, *MAX* is 1 if all of the singularities are ordinary.

[3]We can remove the homogeneous z-variable by setting it to 1, since we are only concerned with finite singularities.

a pair of polynomials of degree at most d in v variables can be computed in $O(d^{2v+1} \log d)$ time [10]. Therefore, X (and Y) can be computed in $O(n^5 \log n + \alpha[n^2])$. Since X and Y are of size $O(n^2)$ and $O(n^2)$ time suffices to evaluate an equation of degree n, the singularities can be computed in $O(n^5 \log n + \alpha[n^2] + n^6)$ time. The flexes, which are the intersections of the curve with its Hessian (Lemma 2.2), can also be computed in $O(\alpha[n^2] + n^6)$ time.

A curve of order n has $O(n^2)$ flexes and singularities (Lemma 2.3). Therefore, it has $O(n^2)$ curve points at its flexes. The bound on the number of singularities is expressed in terms of the maximum number of double points: a curve of order n can have at most $\frac{(n-1)(n-2)}{2}$ double points, and a singularity of multiplicity t counts as $\frac{t(t-1)}{2}$ double points. Since $O(2t)$ pseudo curve points are created at a singularity of multiplicity t, $\frac{2t}{\frac{t(t-1)}{2}} = \frac{4}{t-1} \leq 4$ pseudo curve points are created per double point. Therefore, there are no more than $4 * \frac{(n-1)(n-2)}{2} = O(n^2)$ pseudo curve points.

Consider the time required to compute the pseudo curve points. It takes $O(d^2)$ time to apply a quadratic transformation or a translation to an equation of degree d. It takes $\alpha[d]$ time to compute the intersections of a curve of order d with the y-axis. During the reduction of a singularity to simple points, each quadratic transformation can double the degree of the curve's equation, since y^i becomes $(xy)^i$. Therefore, the equation can become of degree $2^{MAX}n$ during the reduction of a singularity. Finally, $O(n^2)$ quadratic transformations are sufficient to reduce all of the singularities (which account for $O(n^2)$ double points) to simple points [4]. We conclude that a (very pessimistic) bound on the time for computing the pseudo curve points is $O(n^2(2^{MAX}n)^2\alpha[2^{MAX}n])$.

Finally, consider the incidental curve points. Since a line intersects a curve of order n at most n times (Theorem B.1), each of the $O(n^2)$ tangents at singularities and flexes can intersect the curve in at most n points. Thus, there are $O(n^3)$ incidental curve points, and they can be computed in $O(n^2\alpha[n])$ time. We conclude that the cell partition has $O(n^3)$ curve points and $O(n^3)$ convex segments.

Consider the time required to compute the partners of all of the curve points. The dominating expense is the computation of the set $S(W_1)$ of Theorem 3.1 for each curve point W_1. It takes $O(k\alpha[n])$ time to compute $S(W_1)$ for a curve point in a cell with k curve points, $O(k^2\alpha[n])$ time to compute $S(W_1)$ for every curve point in a cell with k curve points, and $O(\sum k_i^2\alpha[n])$ time to compute $S(W_1)$ for every curve point in every cell, where k_i is the number of curve points in cell C_i, and the sum is over all cells C_i. Since $\sum k_i = O(n^3)$, $O(\sum k_i^2\alpha[n]) = O(n^6\alpha[n])$. Therefore, partner computation takes $O(n^6\alpha[n])$ time. (This is another example of an unrealistically pessimistic worst case: a typical curve point will not lie on the boundary of a multisegment cell and its partner will be computed in constant time.) We conclude that preprocessing takes $O(\alpha[n^2] + n^6 + n^2\alpha[2^{MAX}n] + n^4 2^{2\ MAX} + n^6\alpha[n])$ time.

The dominating expense of the actual sorting is the determination of the convex segment that each sortpoint lies on. In the worst case, it requires $O(k\alpha[n]) = O(n^3\alpha[n])$ time to compute the set $S(x)$ of Theorem 3.2 for a sortpoint in a cell with k curve points, and thus $O(mn^3\alpha[n])$ time for all sortpoints. The sorting of p points on a convex segment takes $O(p)$ time (Theorem 2.1). Therefore, in the worst case, the convex-segment method requires $O(mn^3\alpha[n])$ time to traverse $O(n^3)$ convex segments and sort the points on these convex segments. ∎

Corollary 4.1 *Let C be a plane curve of order n. If C has no extraordinary singularities and its cell partition contains no multisegment cells, then m points of C can be sorted in* $O(m + n^3)$ *time, with* $O(n^6 + n^2\alpha[n] + \alpha[n^2])$ *preprocessing.*

Corollary 4.2 *m points on a convex segment of a plane curve of order n can be sorted in O(m) time, without preprocessing.*

4.2 Empirical Results

This section presents execution times for the sorting of some representative curves by the convex-segment and parameterization methods. These empirical results are a good complement to the complexity analysis of Section 1, since they capture the expected case, rather than the worst case, behaviour of the methods.

We do not consider the time required to find a parameterization of the curve or to find the flexes and singularities of the curve. The computation of a curve's parameterization is of approximately the same complexity as the computation of a curve's singularities and flexes, so our comparison of sorting methods should not be biased. Moreover, each of these computations is a preprocessing step that is entirely independent of sorting, and the parameterization, singularities, and flexes of a curve will (or should) often be computed already.

Our results are execution times in seconds on a Symbolics Lisp Machine, and the time spent in disk faults and garbage collection is not included. The source code is written in Common Lisp. The preprocessing time for the convex-segment method is the time required to create the cell partition and find the partners of all of the curve points. The preprocessing and sort times for the convex-segment method are the average of twelve trials, while the sort times for the parameterization method are the average of three trials.

We consider five examples: two rational cubic curves and three non-rational quartic curves. Our first example illustrates the superiority of the convex-segment method. Even when the preprocessing time is added to the sort time, the convex-segment method solves this problem more efficiently than the parameterization method. The convex-segment method's rate of growth is also much smaller. The inferiority of the crawling method is obvious from this example, and we do not consider it further.

Example 4.2.1 *A semi-cubical parabola*

Equation of the curve: $27y^2 - 2x^3 = 0$

Preprocessing time: 0.27 seconds

Parameterization: $\{x(t) = 6t^2, y(t) = 4t^3 \; : \; -\infty < t < +\infty\}$

number of sortpoints	*1*	*2*	*6*
convex-segment	*.01*	*.03*	*.03*
convex-segment + preprocessing	*.28*	*.30*	*.30*
parameterization	*.47*	*.63*	*1.04*
crawling	*3.14*	*2.89*	*4.77*

The second example illustrates that there is sometimes a tradeoff between the convex-segment method (a very fast sort that requires prepro-

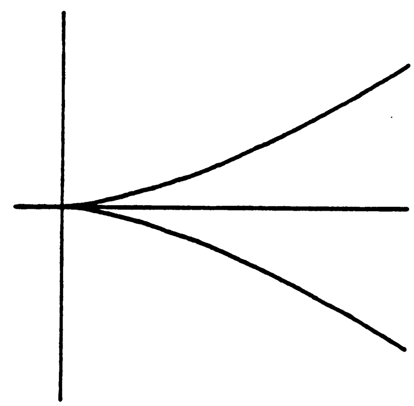


Figure 4.1: Semi-cubical Parabola

cessing) and the parameterization method (a moderately fast sort that does not require preprocessing).

Example 4.2.2 *Folium of Descartes*

Equation of the curve: $x^3 + y^3 - 15xy = 0$

Preprocessing time: 2.81 seconds

Parameterization: $\{x(t) = \frac{15t}{1+t^3},\ y(t) = \frac{15t^2}{1+t^3}\ :\ -\infty\ <\ t\ <\ +\infty\}$

number of sortpoints	*1*	*2*	*5*	*9*
convex-segment	*0.01*	*0.01*	*0.05*	*0.04*
convex-segment + preprocessing	*2.82*	*2.82*	*2.85*	*2.85*
parameterization	*1.01*	*1.07*	*1.76*	*3.17*

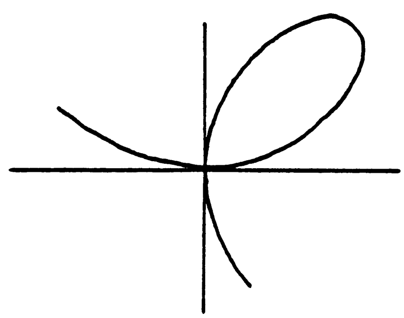


Figure 4.2: Folium of Descartes

The remaining three curves are non-rational, so they are only sorted with the convex-segment method.

Example 4.2.3 *Devil's Curve (with several connected components)*

Equation of the curve: $y^4 - 4y^2 - x^4 + 9x^2 = 0$

Preprocessing time: 2.20 seconds

number of sortpoints	*1*	*4*	*7*
convex-segment	*0.09*	*0.09*	*0.10*
convex-segment + preprocessing	*2.29*	*2.29*	*2.30*

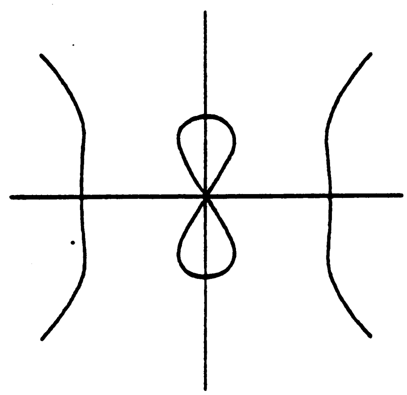


Figure 4.3: Devil's curve

Example 4.2.4 *Limacon*

Equation of the curve: $x^4 + y^4 + 2x^2y^2 - 12x^3 - 12xy^2 + 27x^2 - 9y^2 = 0$

Preprocessing time: 4.62 seconds

number of sortpoints	*2*	*5*	*8*
convex-segment	*.09*	*.30*	*.55*
convex-segment + preprocessing	*4.70*	*4.92*	*5.17*

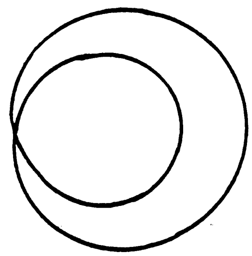


Figure 4.4: Limacon

Example 4.2.5 *Cassinian oval*

Equation of the curve: $x^4 + y^4 + 2x^2y^2 + 50y^2 - 50x^2 - 671 = 0$

Preprocessing time: 5.36 seconds

number of sortpoints	*2*	*4*	*6*
convex-segment	*.14*	*.17*	*.19*
convex-segment + preprocessing	*5.50*	*5.53*	*5.55*

4.3 The Superiority of the Convex-Segment Method

Section 1.3 established that certain curves cannot, or should not, be sorted by the parameterization method: curves that do not possess a rational parameterization and curves for which a rational parameterization cannot

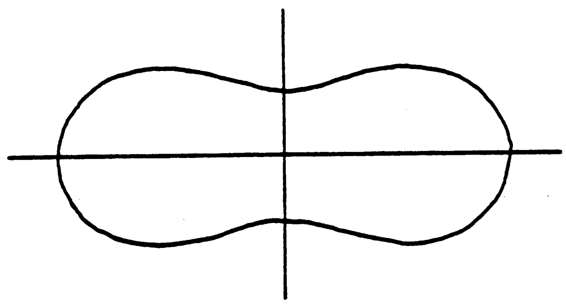


Figure 4.5: Cassinian oval

be efficiently obtained. Therefore, the convex-segment method is often the only viable way to sort points along a curve.

For those curves that can be sorted in either way, the convex-segment method is generally far more efficient than the parameterization method at the actual sorting of the points. However, the parameterization method does not have the expense of preprocessing that the convex-segment method does. Therefore, when only a few points need to be sorted (over the entire lifetime of the curve) and the sorting of these points must be done soon after the definition of the (rational) curve, the parameterization method will usually be the method of choice. The expense of preprocessing will be warranted when sorting time is a valuable resource, as in a real-time application, or when the number of points that will be sorted is large. The convex-segment method will also be preferable when the curve is defined long before it is ever sorted (as with a complex solid model that requires several days, weeks, or even months to develop), since the preprocessing can be done at any time that processing time becomes available before the sort.

We conclude that the convex-segment method is an effective new method for sorting points along an algebraic curve, and that in many situations it is either the only or the best method.

Chapter 5

Applications, Future Work, and Conclusions

This chapter creates a context for sorting within the area of geometric modeling. Section 1 cites several applications of sorting, thereby establishing its importance. Section 2 discusses various research problems that are suggested by our work on sorting, and we end with some conclusions.

5.1 Applications

The sorting of points along an algebraic curve has many applications in geometric modeling. The components of a geometric model are faces, edges, and vertices, which are represented by patches of algebraic surfaces, segments of algebraic curves, and points, respectively. The sorting of points is a problem that arises naturally in the manipulation of these components. We consider a number of applications.

Problem 1 Given a set S of points on an algebraic curve C, determine the points of S that lie on an edge E of C.

Solution E is defined by the implicit equation of C and two endpoints V_1 and V_2. If S is sorted along $\widehat{V_1V_2}$, then the points of S that do not lie on the edge will be ignored and the sorted list will only contain the points of S that lie on the edge.

Problem (1) is a very basic problem in geometric modeling. It must be solved regularly in operations ranging from intersection to display. We offer two examples of its use.

Problem 2 Compute the intersection of two edges.

Solution Let E_1 and E_2 be edges of the curves C_1 and C_2, respectively. Once the points of intersection of the two curves have been computed (perhaps by resultants), Problem (1) can be applied (twice) to determine the points of $C_1 \cap C_2$ that lie on $E_1 \cap E_2$, since $E_1 \cap E_2 = [(C_1 \cap C_2) \cap E_1] \cap E_2$.

Problem 3 Determine a bounding box for an edge E.

Solution In motion planning, it is useful to approximate a geometric model by a simple superset, because this makes interference detection simpler. The more expensive interference detection with the geometric model can be reserved for those situations when the solid approaches close enough to an obstacle that interference is detected with the simple superset. Bounding regions are also useful for problems such as (1) above, for they allow points that clearly do not satisfy a condition to be quickly discarded. We can define a bounding rectangle for an edge E by the minimum and maximum x and y values of E. Consider the computation of the maximum x value of E. (The other extrema are computed in a similar manner.) E's x-maximum is either attained at a local x-maximum of E's curve or at an endpoint of the edge. Therefore, in order to determine the maximum x-value of E, the local x-maxima of the curve must be computed (as solutions of $f_y = f = 0$, where f is the implicit equation of E's curve), and then restricted to the subset that lies on E. This restriction is an instance of Problem (1).

Problem 4 Determine if a point lies within a piecewise-algebraic plane patch.

Solution This problem is fundamental to the display of a geometric model. A piecewise-algebraic plane patch is defined by a closed boundary consisting of a simply-connected collection of plane algebraic curve segments. The problem of determining whether a point Q lies within the closed boundary reduces to the problem of sorting points by the following mapping. Consider the straight line L defined by a vertex V on the boundary and the point Q. We compute the set $\mathcal{I}$ of intersections of L with the algebraic curve segments of the patch's boundary, through several applications of Problem (2). The points of $\mathcal{I}$ and the point Q are then sorted along the line L. By applying the Jordan curve theorem, the points of $\mathcal{I}$ can be grouped into pairs, and inside/outside intervals can be determined. Q lies within the patch if and only if it lies on an inside interval.

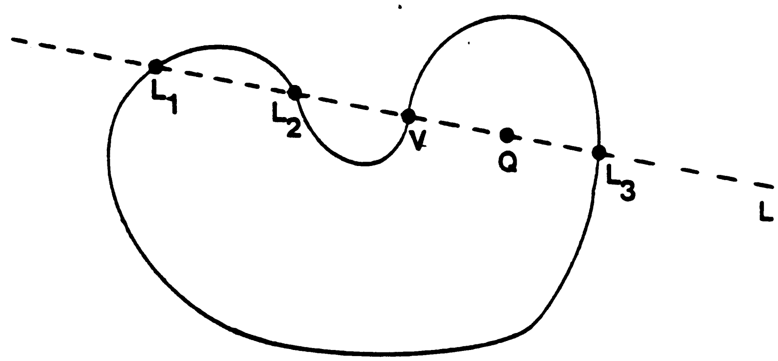


Figure 5.1: Deciding if Q lies inside the plane patch

Example 5.1.1 *Consider the plane patch of Figure 5.1.* $\mathcal{I} = \{L_1, L_2, L_3, V\}$*. The sorted order of* $\mathcal{I} \cup \{Q\}$ *is* L_1, L_2, V, Q, L_3*. Therefore, the intervals of* L *that are inside the patch are* $\overline{L_1L_2}$ *and* $\overline{VL_3}$*. Since* Q *lies on one of these* *inside intervals* $(\overline{VL_3})$*, it lies inside of the patch.*

Problem 5 Determine if a point lies within a piecewise-algebraic convex surface patch.

Solution A piecewise-algebraic surface patch is defined by a closed, simply-connected loop of boundary edges on a primary surface F. The edges are algebraic space curve segments defined by the intersection of secondary surfaces G_i with the primary surface F. This problem is a direct extension of Problem (4), with one exception. Instead of using a line L defined by the point Q and a vertex on the boundary, we use a planar cross-section of the primary surface, where the plane is defined by Q and two vertices on the boundary. The primary surface is assumed to be convex in order to guarantee that the planar cross-section is a connected curve and that a sort of points along the cross-section is therefore well-defined. Since we are now sorting points along a curve rather than a line, nontrivial sorting is required both to find the points of intersection with the patch boundary and to sort them.

Problem 6 Compute the intersection of two solid models. (See Section 1.1.)

These six problems give an indication of the importance of sorting points along an algebraic curve. They also reveal that there are essentially two ways in which sorting can be used: Problems (1)-(3) use sorting as a means of restricting a set of points to a specific subset, while Problems (4)-(6) use sorting as a means of introducing an even-odd parity to a set of points.

5.2 Future Work

5.2.1 Parameterization

Our investigation of sorting has revealed some problems that require further attention. One of the most obvious areas for future research is the parameterization of surfaces of higher degree. Appendix C presents some methods for parameterizing surfaces of degree two and three, but we know of no practical methods for higher degrees.

5.2.2 Curves with Several Connected Components

Curves with several connected components are more challenging than curves with a single connected component.[1] Example 5.2.1 illustrates that all of the sortpoints must lie on the same connected component when the convex-segment method is used to sort points on a connected component that has

[1] Just as multisegment cells are more challenging than cells with a single convex segment.

no curve points. This requirement is not necessary (i.e., the sortpoints can be strewn over several connected components) if the points are being sorted on a connected component CC that contains at least one curve point, since a convex-segment traversal is then possible. Any sortpoints that do not lie on CC will be ignored (using Theorem 3.2) in the same way that any sortpoints on CC that do not lie on the sort segment are ignored.

Example 5.2.1 *The quartic curve of Figure 5.2 has four connected components, but no flexes or singularities. Thus, this curve has no walls in its cell partition, and it will appear as if all of the sortpoints lie on the same convex segment. This will not be a correct conclusion unless all of the sortpoints lie on the same connected component.*

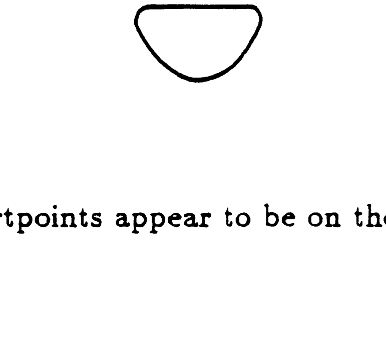


Figure 5.2: These sortpoints appear to be on the same convex segment

Unfortunately, the points that are to be sorted may lie on several connected components. For example, the sortpoints will often be generated by intersecting a curve with another curve or surface (as evidenced by the previous section), and the resulting points may be spread over several connected components. Therefore, before sorting can proceed, those sortpoints that lie on connected components with no curve points must be divided into connected components. The most obvious way of doing this is to create a boundary about each connected component (Figure 5.3).

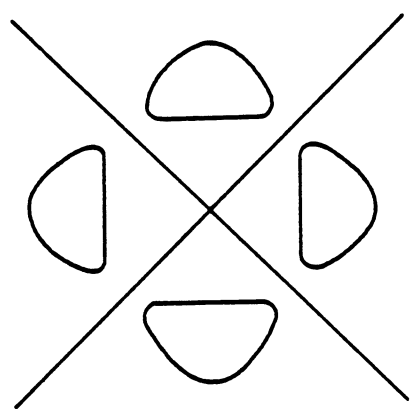


Figure 5.3: Component separation of a quartic curve

The creation of these boundaries turns out to be simple for cubic curves, however a general solution may be very difficult.[2] Collins' cylindrical algebraic decomposition provides a possible solution. This decomposition can

[2]Hilbert's 16^{th} problem is to determine the relative position of the connected components of a nonsingular algebraic plane curve [22].

be used to determine the topology of an algebraic curve [6,23], from which it should be straightforward to determine boundaries for each connected component. Although the computation of a cylindrical algebraic decomposition of $\Re^d$ is double-exponential (or parallel-exponential) in d, we are only concerned with decompositions of the plane, so $d = 2$ and the method may be tractable.

The sorting of points along a curve of several connected components is also difficult when the parameterization method of sorting is used. The following questions arise: (1) should each connected component have a separate parameterization? (2) if so, how does the implicit equation of a curve produce several, independent parameterizations? (3) if not, how can a single parameterization be split over several connected components (i.e., how can the range of parameter values that is associated with each connected component be determined)?

5.2.3 A Theory of Finite Precision

The algorithms of geometric modeling must be implemented on a computer of finite precision. This can cause instabilities that, because of their geometric nature, are quite different from the instabilities studied in conventional numerical analysis. For example, in the creation of a cell partition, the tangent of a flex is intersected with the curve. Since the curve can be very flat about a flex, a small error in the tangent will cause the tangent to have more than one intersection with the curve near the flex. Although this particular problem is easily solved (by merging the intersections), finite precision problems may eventually impede progress in other applications. Therefore, the development of a systematic theory for the resolution of geometric problems arising from finite precision would be an important contribution to geometric modeling.

5.2.4 The Importance of Flexes, Singularities, and Projections

The convex-segment method has revealed the importance of the singularities and flexes of a curve. Singularities are also used to develop a curve's parameterization (Section 1.3.1). It may be advisable to include flexes and singularities as an integral component of a curve's representation in a geometric model. (For a space curve, the singularities and flexes of its projection would be stored.) Certainly, the discovery of more efficient algorithms for the computation of singularities and flexes would be a significant contribution.

Our investigation of sorting has also demonstrated that the best way to solve a problem for a space curve may be to reduce the dimension, and hopefully the complexity, of the problem by translating it to an equivalent problem for a projection of the space curve.

5.3 Conclusions

We have developed a new method of sorting points along an algebraic curve that is superior to the conventional methods of sorting. Many curves that could not be sorted, or that could only be sorted slowly, can now be sorted efficiently. The development of our new method has also illustrated how an algebraic curve can be decomposed into convex segments, and how the ambiguity of sorting through a singularity can be resolved.

The creation and manipulation of curves and surfaces is of major importance to geometric modeling. A sophisticated geometric modeling system should offer a rich collection of tools to aid this manipulation. Our work on sorting has been an attempt to develop one of these tools. The progress of geometric modeling depends upon the development of more tools and upon the extension of more computational geometry algorithms from polygons to curves and surfaces of higher degree.

Appendix A

Definitions

The primary sources for the definitions of this appendix are Lawrence [27] and Walker [35].

An **algebraic plane curve** is the zero set of a bivariate polynomial of positive degree over a field K (in our case, $K = \Re$). That is, a typical algebraic plane curve is { (x,y) | F(x,y) = 0 and F(x,y) is a polynomial of degree $n > 0$ in x,y with coefficients in a field K}. The **order** of an algebraic plane curve is the degree of its defining polynomial. A **conic** is a plane curve of order two.

A **space curve** is a curve that does not lie in a plane. An **algebraic space curve** is the intersection of two surfaces, each of which is represented by a trivariate polynomial of positive degree over a field K.

The **implicit representation** of a curve or surface is its representation in terms of the zero set of a system of equations.

Let P be a point of the curve $f(x, y) = 0$. Suppose that all derivatives of f up to and including the $r - 1^{st}$ vanish at P, but that at least one r^{th} derivative does not vanish at P. P is called a point of **multiplicity** r. Every line through P has at least r intersections with the curve at P, and precisely r such lines, properly counted, have more than r intersections. The exceptional lines are called the **tangents** to the curve at P.

A **singularity** (or **singular** point) is a point of multiplicity two or more. A singularity is a point where two different branches of the same connected component of a curve touch or a point where the curve changes direction sharply. A **simple** (resp., **double**) point is a point of multiplicity one (resp., two). A singularity of multiplicity r is **ordinary** if its r tangents are distinct. A singularity is **extraordinary**[1] if it is not ordinary (Figure A.1(c)). A **node** is an ordinary double point, and a **cusp** is an extraordinary double point (Figure A.1). A segment is **nonsingular** if it does not contain any singularities.

Any polynomial $F(x, y)$ of degree n has a factorization $F = F_1 F_2 \ldots F_r$ into irreducible polynomials, unique to within constant multiples. The curves $F_1(x, y) = 0, \ldots, F_r(x, y) = 0$ are called the **irreducible components** of the curve $F(x, y) = 0$. An **irreducible** curve is a curve with one

[1]This term is not from the literature.

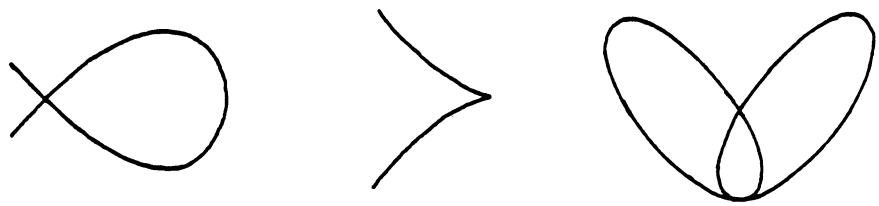


Figure A.1: (a) node (b) cusp (c) extraordinary singularity

irreducible component.

Two points of a curve are **connected** if they can be joined by a continuous path on the curve. A **connected component** of a curve is a maximal subset of the curve such that any two points of the subset are connected. A connected component of an algebraic curve is either unbounded or it forms a closed cycle. A curve is **closed** if all of its connected components are closed cycles, otherwise it is **open**.

$g : I \subseteq \Re \rightarrow \Re^2$ is a **parameterization** of the plane curve C if it puts the points of the curve into an almost one-to-one correspondence with the points of a line segment, by expressing the coordinates of the curve independently as functions of a single variable t: x=j(t), y=k(t). More formally, $g(t) = (x(t),y(t)) : I \subseteq \Re \rightarrow \Re^2$ is a parameterization of the plane curve f(x,y) = 0 if and only if

(i) with only a finite number of exceptions, if $t_0 \in I$, then f(x(t_0),y(t_0)) = 0 (i.e., almost all of g(I) is contained in the curve); and

(ii) with only a finite number of exceptions, if (x_0,y_0) is a point of the curve, then there is a unique $t_0 \in I$ such that x_0 = x(t_0), y_0 = y(t_0) (i.e., g is one-to-one and onto almost everywhere).

The definition of a parameterization g(t) = (x(t),y(t),z(t)) of a space curve is similar.

A function f(x) is **rational** if it can be expressed as the ratio of two polynomials: $f(x) = \frac{g(x)}{h(x)}$. A parameterization (x(t),y(t)) is **rational** if both x(t) and y(t) are rational.

Let

$$f(x_1, \ldots, x_r) = a_0 + a_1 x_r + \ldots + a_n x_r^n$$

$$g(x_1, \ldots, x_r) = b_0 + b_1 x_r + \ldots + b_m x_r^m$$

where $a_i, b_i \in \Re[x_1, \ldots, x_{r-1}]$, $a_n b_m \neq 0$, and $n, m > 0$. The **resultant** of f and g with respect to x_r is

$$R(x_1,\ldots,x_{r-1}) = \begin{vmatrix} a_0 & a_1 & \ldots & a_n & & \\ & a_0 & \ldots & a_{n-1} & a_n & \\ & & \ldots & & & \\ & & & a_0 & \ldots & a_n \\ b_0 & b_1 & \ldots & b_m & & \\ & \ldots & \ldots & & & \\ & & & b_0 & \ldots & b_m \end{vmatrix}$$

where there are m rows of a's and n rows of b's, the rows being filled out by zeros.

Let $\widehat{W_1W_2}$ be a convex segment. The **inside** of the chord $\overleftrightarrow{W_1W_2}$ is the halfplane that contains $\widehat{W_1W_2}$. The inside of the chord $\overleftrightarrow{W_1W_2}$ is found by crawling from W_1 to a point on $\widehat{W_1W_2}$ and determining the side of $\overleftrightarrow{W_1W_2}$ that this point lies on. The computation of the inside of the chord of each convex segment that lies in a multisegment cell of the cell partition is a preprocessing step.

Appendix B

Lemmas

This appendix presents various lemmas and theorems that are important to the development of the theory of sorting. Due to their technical nature, we find it more convenient to place these results in an appendix.

Theorem B.1 (Bezout's Theorem [35, p. 59]) *If two algebraic plane curves, of orders m and n, have more than mn common points, then they have a common component.*

Lemma B.1 ([35, pp. 25, 26, 59]) *Let $R(x_1, \ldots, x_{r-1})$ be the resultant of $f(x_1, \ldots, x_r)$ and $g(x_1, \ldots, x_r)$ with respect to x_r. $R(\alpha_1, \alpha_2, \ldots, \alpha_{r-1}) = 0$ if and only if there exists α_r such that $f(\alpha_1, \ldots, \alpha_r) = g(\alpha_1, \ldots, \alpha_r) = 0$.*

Lemma B.2 *The tangent of an algebraic curve changes continuously on nonsingular segments of the curve.*

Proof Since all polynomial functions are continuous, this is a corollary of Lemma 2.4. ▮

Definition B.1 *A polygon is* **simple** *if (i) any pair of edges of the polygon are either disjoint or intersect only at their endpoints, and (ii) no more than two edges intersect at each endpoint.*

The two common definitions for convexity of a polygon P are:

- if $v, w \in P$, then $\{tv + (1-t)w \mid 0 \leq t \leq 1\} \subset P$
- $P = \{\sum_{i=1}^{n} \lambda_i v_i \mid \sum_{i=1}^{n} \lambda_i = 1,\ 0 \leq \lambda_i \leq 1\}$, where $v_1, \ldots, v_n$ are the vertices of P.

We present an alternative characterization of convexity that works with the boundary rather than the interior of the polygon.

Lemma B.3 *Let $P = v_1 \ldots v_n$ be a simple polygon. P is convex if and only if, for every edge $E = \overline{v_i v_j}$, the line $\overleftrightarrow{v_i v_j}$ does not intersect $P \setminus E$.*

Proof $\Rightarrow$: Assume that P is convex. Suppose, for the sake of contradiction, that $\overline{v_i v_j}$ is an edge of P such that the line $\overleftrightarrow{v_i v_j}$ intersects $P \setminus E$. Let w be the first such intersection and assume, without loss of generality, that w lies on the ray $\overrightarrow{v_i v_j}$. Then there exists $x \in \overline{v_j w}$ such that $x \notin P$, which contradicts the first definition of polygonal convexity.

$\Leftarrow$: Assume that, for every edge E = $\overline{v_i v_j}$, the line $\overleftrightarrow{v_i v_j}$ does not intersect

$P \setminus E$. Suppose, for the sake of contradiction, that there exists $v, w \in P$ and $t \in [0,1]$ such that $X = tv + (1-t)w \notin P$ (Figure B.1). Since $v \in P$ and $X \notin P$, $\overline{vX} \cup \{v\}$ must cross the boundary of P, say at e on edge E. v (or, if $v = e$, the polygon in the neighbourhood of v) and w lie on opposite sides of the line defined by E. Therefore, since $v, w \in P$, some edge of P must cross over E's line, say at Y. Since P is a simple polygon, Y cannot be a point of E. Thus, the continuation of E intersects $P \setminus E$, a contradiction. Therefore, for all $v, w \in P$ and $t \in [0,1]$, $tv + (1-t)w \in P$, and P is convex. ∎

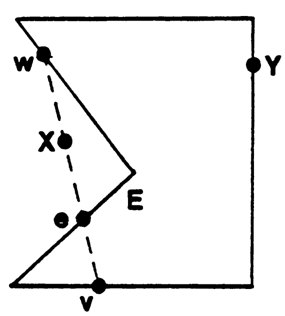


Figure B.1: An example for Lemma B.3

Recall from Section 2.2 what it means for a line to cross a curve.

Lemma B.4 *A plane curve C crosses a line L at a nonsingular point P if and only if the number of intersections of L with C at P is odd.*

Proof Assume, without loss of generality, that L is the x-axis and P is the origin. Let $N_\epsilon(0)$ be a small segment of the curve near the origin: i.e.,

$N_\epsilon(0) = \{ c \in C \mid \text{dist}_C(c, 0) < \epsilon \}$, where $\text{dist}_C(c_1, c_2)$ is the length of the curve segment between two points c_1, c_2 of C. Since the directed tangent to the nonsingular origin is ($\pm$1,0) and the tangent to an algebraic curve changes continuously on nonsingular segments (Lemma B.2), there exists $\epsilon > 0$ such that no two points of $N_\epsilon(0)$ have the same abscissa. Within this neighbourhood, the curve can be represented by a function $y = g(x)$. We expand $g(x)$ into a Taylor series:

$$g(x) = g(0) + g'(0)x + g''(0)x^2 + \dots \tag{B.1}$$

Since the lowest order term in (B.1) dominates all other terms as $x \to 0$, $g(x)$ changes sign as x changes sign if and only if $\min\{ i \mid g^{(i)}(0) \neq 0\}$ is odd. Also, $g(x)$ changes sign as x changes sign if and only if C crosses L (the x-axis) at P (the origin). Therefore, C crosses L at P if and only if $\min\{ i \mid g^{(i)}(0) \neq 0\}$ is odd.

Since $y - g(x) = 0$ represents the curve C near the origin and $x = t,\ y = 0$ is a parameterization of L, the intersections of L with C near the origin are associated with the roots of $g(t) = 0$. In particular, using (B.1) above, the multiplicity of the intersection of L with C at the origin is $\min\{ i \mid g^{(i)}(0) \neq 0\}$. Therefore, C crosses L at P if and only if the multiplicity of intersection of L with C at P is odd. ∎

Lemma B.5 *An infinite segment of an algebraic curve cannot be bounded within a closed region.*

Proof Let S be an infinite segment of an algebraic curve and suppose that S lies in a closed region. Since S must twist infinitely often to avoid crossing the boundaries of the region, a line can be found that intersects S in an arbitrarily large number of points, contradicting Bezout's Theorem (Theorem B.1). ∎

Lemma B.6 *Let $\widehat{PQ}$ be a nonconvex segment of a curve F such that $\widehat{PQ}$ contains no singularities or floxes. There exists a line L that crosses $\widehat{PQ}$ in at least three distinct points.*

Proof Since $\widehat{PQ}$ is nonconvex, there exists a line $\mathcal{L}$ that intersects $\widehat{PQ}$ at least three times, such that not all of the intersections occur at a flex of even order. There are three cases to consider.

Case 1 If $\mathcal{L}$ crosses $\widehat{PQ}$ in at least three distinct points, then we are done.

Case 2 Suppose that $\mathcal{L}$ intersects $\widehat{PQ}$ at less than three distinct points. Since $\widehat{PQ}$ contains no floxes, $\mathcal{L}$ must intersect $\widehat{PQ}$ at two distinct points. By the pigeon-hole principle, two of the intersections must occur at the same point. That is, $\mathcal{L}$ is tangent to the curve at some point.

Case 3 Suppose that $\mathcal{L}$ intersects $\widehat{PQ}$ at three or more distinct points but crosses $\widehat{PQ}$ at less than three points. Thus, $\mathcal{L}$

touches but does not cross $\widehat{PQ}$ at some point x. Since $\widehat{PQ}$ is nonsingular, the tangent changes continuously, so $\mathcal{L}$ must be tangent to $\widehat{PQ}$ at x in order to touch but not cross $\widehat{PQ}$.

Therefore, either $\mathcal{L}$ already satisfies the requirements or there exists $x \in \widehat{PQ}$ such that x's tangent T_x strikes $\widehat{PQ}$ at another point (Figure B.2). Let $y \neq x$ be an intersection of T_x with $\widehat{PQ}$ such that $\widehat{xy}$ lies strictly inside T_x (i.e., y is the closest intersection to x). For any $\epsilon > 0$, let L_ϵ be the line such that (i) L_ϵ is parallel to T_x, (ii) L_ϵ lies inside T_x, and (iii) L_ϵ is at a distance of ϵ from T_x. It can easily be shown that there exists an $\epsilon > 0$ such that L_ϵ crosses the curve at least three times. ∎

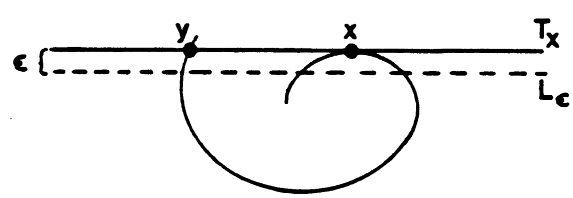


Figure B.2: Some L_ϵ will cross the curve at least three times

Lemma B.7 *If $\widehat{W_1 W_2}$ is a convex segment in the cell C, then W_1 and W_2 face each other (with respect to C).*

Proof If W_1 is not a flox, then W_1 faces W_2 simply because W_2 must lie inside of W_1's tangent by convexity.

Suppose that W_1 is a flox and W_2 lies on W_1's tangent (Figure B.3). The convexity of $\widehat{W_1W_2}$ implies that a point of $\widehat{W_1W_2}$ lies inside the tangent of any other point of $\widehat{W_1W_2}$. Therefore, W_2 will lie strictly inside of the curve's tangent as the curve leaves W_1 along $\widehat{W_1W_2}$. That is, W_2 and the outside endpoint of W_1's cell segment with respect to C will lie on opposite sides of W_1. Therefore, W_1 faces W_2 with respect to C.

By symmetry, W_2 faces W_1 (with respect to C). ∎

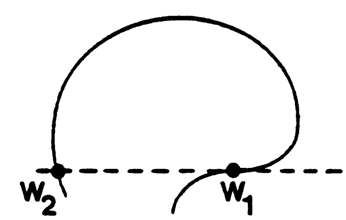


Figure B.3: W_2 lies on W_1's tangent

Lemma B.8 *Let W_1 and W_2 be partners. If W_2 lies on W_1's tangent, then W_1 must be a flox. (Thus, if W_2 lies on W_1's tangent, then W_2 must lie on W_1's wall.)*

Proof Assume that W_2 lies on W_1's tangent. Suppose that W_1 is an incidental curve point. Since $\widehat{W_1W_2}$ is convex, it must look like Figure B.4(a). Thus, since W_1's tangent is not the same as W_1's wall, W_1's wall must cross $\widehat{W_1W_2}$ and split $\widehat{W_1W_2}$ over two cells, which is a contradiction.

Suppose that W_1 is a pseudo curve point. Let V be the singularity from which W_1 was derived. Either the singularity's tangents cross $\widehat{W_1W_2}$ (Figure B.4(b)), a contradiction as in the incidental case, or $\widehat{W_1W_2}$ intersects $\widehat{VW_1}$ (Figure B.4(c)), causing a singularity in the interior of the cell, which is also a contradiction (since singularities will only occur on the boundaries of a cell, by the definition of a cell partition). ∎

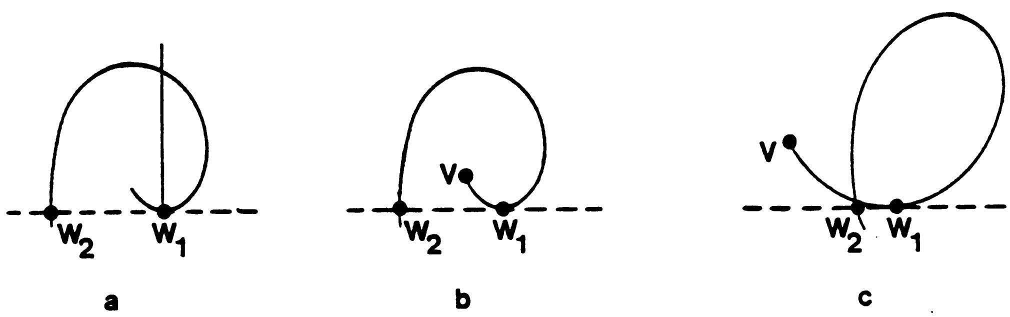


Figure B.4: (a) incidental W_1 (b-c) pseudo W_1

Lemma B.9 *Let x be a point of a curve such that the tangent of x strikes the curve again at y. Let $z \in \overleftrightarrow{xy} \setminus \overrightarrow{xy}$. Suppose that y lies strictly outside of the curve's tangent as the curve leaves x along $\widehat{xy}$ (Figure B.5(a)). Then $\widehat{xy}$ must contain a flox, a singularity, or a point of $\overrightarrow{xz}$.*

Proof Suppose that $\widehat{xy}$ does not contain a flox or a singularity. Thus, the curve cannot cross itself on $\widehat{xy}$ and there cannot be any point on $\widehat{xy}$ where the curvature changes from concave to convex or vice versa. In other

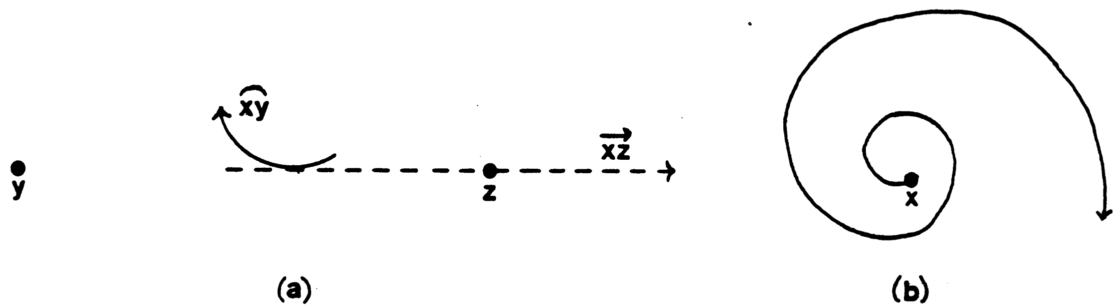


Figure B.5: (a) x, y, and z in Lemma B.9 (b) $\widehat{xy}$ without singularities or floxes

words, $\widehat{xy}$ must spiral around in ever larger circles (Figure B.5(b)). Since y lies strictly outside of the curve's tangent as the curve leaves x along $\widehat{xy}$, the curve must spiral by an angle of at least 2π to get from x to y. Therefore, $\widehat{xy}$ must cross $\vec{xz}$. ∎

Appendix C

Parameterization Algorithms

This appendix considers several algorithms that have been developed for the parameterization of plane curves and surfaces of degree two and three. The first method proceeds by linearizing the implicit equation of the curve or surface with respect to one of its variables [2,3]. For example, to parameterize a degree two curve, a linear transformation of the form

$$X = \frac{a_1 x + b_1 y + c_1}{a_3 x + b_3 y + c_3}, \quad Y = \frac{a_2 x + b_2 y + c_2}{a_3 x + b_3 y + c_3}$$

is applied to the equation of the curve, where b_1, b_2, and b_3 are chosen so that the y^2 term is eliminated, and a_i and c_i are chosen so as to make the transformation well-defined (by ensuring that the matrix

$$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}$$

is nonsingular) [2]. The resulting equation is linear in y and easy to parameterize.

Example C.0.1 *Consider the parameterization of the circle $x^2 + y^2 - 1 = 0$. The implicit equation is linearized with respect to y by applying the transformation $x = \frac{x_n + y_n}{y_n}$, $y = \frac{1}{y_n}$:*

$$\begin{aligned} x^2 + y^2 - 1 = 0 \quad & \rightarrow (\tfrac{x_n + y_n}{y_n})^2 + \tfrac{1}{y_n^2} - 1 = 0 \\ & \equiv \tfrac{x_n^2 + y_n^2 + 2x_n y_n + 1 - y_n^2}{y_n^2} = 0 \\ & \equiv x_n^2 + 2x_n y_n + 1 = 0 \end{aligned}$$

This equation is simple to parameterize, because we can solve for y_n in terms of x_n ($y_n = \frac{-x_n^2 - 1}{2x_n}$), yielding the parameterization $x_n = t$, $y_n = \frac{-t^2-1}{2t}$. A parameterization of the original circle is found by substituting back into the transformation equations:

$$x = \frac{x_n}{y_n} + 1 = \frac{t(2t)}{-t^2 - 1} + \frac{-t^2 - 1}{-t^2 - 1} = \frac{t^2 - 1}{-t^2 - 1} = \frac{1 - t^2}{1 + t^2}$$

$$y = \frac{1}{y_n} = \frac{-2t}{1 + t^2}$$

The parameterization of plane curves of degree three and of surfaces of degree two and three by the linearizing technique is analogous. The linearizing technique becomes impractically slow for degrees four and five [7], and it does not generalize to higher degrees (because of the lack of a general formula for the solution of equations of degree five or more [16]).

Another method for parameterizing curves of degree two is to solve for the variables in a template parameterization [19]. Since a plane curve of order two can be parameterized in homogeneous coordinates by four polynomials of degree two [2], a template for the parameterization can be created:

$$x(t) = \frac{a * t^2 + b * t + c}{d * t^2 + e * t + f}$$

$$y(t) = \frac{g * t^2 + h * t + j}{k * t^2 + l * t + m}$$

Two points of the curve and the tangents at these points are assumed to be known. These points and tangents, along with some other conditions, are used to solve for the variables in the template parameterization by substituting into the equation of the curve. It is not clear whether this technique can be generalized to higher degrees, but the preliminary evidence is not encouraging.

A surface of degree two can be parameterized by normalizing the surface's equation to one of a number of forms for which a parameterization is already known, such as $x^2 + y^2 - 1 = 0$ if the surface is an elliptic cylinder [28]. This technique collapses for surfaces of higher degree, since no exhaustive classification, and thus no class of normal forms, is available for these surfaces.

Bibliography

[1] Abhyankar, S.S. *Historical Ramblings in Algebraic Geometry and Related Algebra.* The American Mathematical Monthly, 83:6, June-July 1976, pp. 409-448.

[2] Abhyankar, S.S. and Bajaj, C. *Automatic Parameterization of Rational Curves and Surfaces I: Conics and Conicoids.* Computer Aided Design, 19:1, 1987, pp. 11-14.

[3] Abhyankar, S.S. and Bajaj, C. *Automatic Parameterization of Rational Curves and Surfaces II: Cubics and Cubicoids.* To appear in Computer Aided Design.

[4] Abhyankar, S.S. and Bajaj, C. *Automatic Parameterization of Rational Curves and Surfaces III: Algebraic Plane Curves.* Technical Report #619, Dept. of Computer Science, Purdue University, August 1986.

[5] Arnon, D.S. *Algorithms for the Geometry of Semi-Algebraic Sets.* Ph.D. dissertation, Technical Report #436, Dept. of Computer Science, University of Wisconsin–Madison, 1981.

[6] Arnon, D.S. *Topologically Reliable Display of Algebraic Curves.* Computer Graphics, 17:3, July 1983, pp. 219-227.

[7] Bajaj, C. Personal communication.

[8] Berlekamp, E.R. *Factoring Polynomials over Large Finite Fields.* Mathematics of Computation, 24:111, July 1970, pp. 713-735.

[9] Bajaj, C., Hoffmann, C., and Hopcroft, J.E. *Tracing Algebraic Curves I: Plane Curves.* Technical Report #637, Dept. of Computer Science, Purdue University.

[10] Collins, G. *The Calculation of Multivariate Polynomial Resultants.* Journal of the ACM, 18:4, October 1971, pp. 515-532.

[11] Collins, G. *Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition.* Second GI Conference on Automata Theory and Formal Languages, Berlin: Springer-Verlag, Lecture Notes in Computer Science, Vol. 33, 1975, pp. 134-183.

[12] Do Carmo, M.P. *Differential Geometry of Curves and Surfaces.* Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1976.

[13] Forsythe, G.E., Malcolm, M.A., and Moler, C.B. *Computer Methods for Mathematical Computations.* Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1977, p. 161.

[14] Fulton, W. *Algebraic Curves: An Introduction to Algebraic Geometry.* Menlo Park, California: Benjamin/Cummings Publishing Co., 1974.

[15] Gordon, V. and Sementsov-Ogievskii, M. *A Course in Descriptive Geometry.* Edited by V. Gordon and transl. by L. Levant. Moscow: Mir Publishers, 1980.

[16] Hernstein, I. *Topics in Algebra.* New York: John Wiley and Sons, 1975.

[17] Hilbert, D. and Cohn-Vossen, S. *Geometry and the Imagination.* Transl. by P. Nemenyi. New York: Chelsea, 1952.

[18] Hoffmann, C. and Hopcroft, J.E. *The Potential Method for Blending Surfaces and Corners.* In *Geometric Modeling*, G. Farin, ed., Philadelphia: SIAM Publishing, 1987.

[19] Hoffmann, C. and Hopcroft, J.E. Internal design document, Cornell Robotics Project.

[20] Hopcroft, J.E. and Krafft, D.B. *The Challenge of Robotics for Computer Science*. Reprint of a chapter of *Advances in Robotics, Vol. 1: Algorithmic and Geometric Aspects of Robotics*, J. Schwartz and C. Yap, eds., Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1987.

[21] Jenkins, M. *Algorithm 493*, ACM Transactions of Mathematical Software, 1:2, June 1975, pp. 178-189.

[22] Kaplansky, I. *Hilbert's Problems: Preliminary Edition.* Lecture Notes in Mathematics, Dept. of Mathematics, University of Chicago, 1977, p. 153.

[23] Kozen, D. and Yap, C. *Algebraic Cell Decomposition in NC.* Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science, October 1985, pp. 515-521.

[24] Kreyszig, E. *Differential Geometry.* Toronto: University of Toronto Press, 1959.

[25] Lang, S. *Introduction to Algebraic Geometry.* Reading, Massachusetts: Addison-Wesley, 1972.

[26] Lang, S. *Linear Algebra.* Reading, Massachusetts: Addison-Wesley, 1971.

[27] Lawrence, J.D. *A Catalog of Special Plane Curves.* New York: Dover Publications, 1972.

[28] Levin, J.Z. *A Parametric Algorithm for Drawing Pictures of Solid Objects Composed of Quadric Surfaces.* CACM, 19:10, October 1976, pp. 555-563.

[29] Levin, J.Z. *Mathematical Models for Determining the Intersections of Quadric Surfaces.* Computer Graphics and Image Processing 11, 1979, pp. 73-87.

[30] Mortenson, M. *Geometric Modeling.* New York: John Wiley, 1985.

[31] Namba, M. *Geometry of Projective Algebraic Curves.* New York: Marcel Dekker, 1984.

[32] Orzech, G., and Orzech, M. *Plane Algebraic Curves: An Introduction via Valuations.* New York: Marcel Dekker, 1981.

[33] Salmon, G. *A Treatise on the Analytic Geometry of Three Dimensions.* London: Longmans, Green and Co., 1915, Volume II.

[34] Sommerville, D.M.Y. *Analytical Geometry of Three Dimensions.* Cambridge: Cambridge University Press, 1934.

[35] Walker, R.J. *Algebraic Curves.* New York: Springer Verlag, 1978.

[36] Wang, P. and Rothschild, L. *Factoring Multivariate Polynomials over the Integers*. Mathematics of Computation, 29:131, pp. 935-950.