Regular Paper

On Sorting Points along a Real Algebraic Curve

JiangQian Ying* and Noboru Sugie*

In this paper we use the Cylindrical Algebraic Decomposition (CAD) method to develop algorithms for sorting points along a real algebraic curve. Our approach has less computational complexity than the convex decomposition approach propose by Johnstone and Bajaj [1]. To sort m points along a real algebraic curve of degree n, the convex decomposition approach costs $O(mn^3\beta[n]+m\log m)$ time, where $\beta[n]$ is the cost of finding the real roots of an algebraic equation of order n. In contrast, our approach costs $O(mn^2+m\alpha[n]+m\log m)$ time, where $\alpha[n]$ is the cost of computing the number of real roots of an order n algebraic equation on a certain interval, which is obviously no more expensive than $\beta[n]$.

We also explore the cyclic structure of real algebraic curves: we show how a curve can be naturally decomposed into a finite number of closed cycles and open chains, and claim that this number is no larger than the genus plus the degree of the curve.

1. Introduction

Before proceeding, we will briefly explain our terminology. In this paper, by an "algebraic curve" we mean an algebraic curve defined over the real number field, that is, a real algebraic curve. However, since in mathematics an "algebraic curve" conventionally means an algebraic curve defined over the complex number field, we sometimes use the term "real algebraic curve" to emphasize the distinction.

Sorting points along an algebraic curve [1] is a very interesting problem. A typical version of this problem is defined as follows: Given an algebraic curve segment with end points A and B, and a finite set of points, all in a plane, how do we find the subset of points lying on the segment and sort these points in the order in which they would be encountered in traveling from A to B along the curve? (Fig. 1).

In [1] this problem is solved by decomposing the whole curve into convex segments (a segment is convex if it lies entirely on the boundary of its convex hull). Let $P_1, \cdots, P_n$ be points on a convex segment, and let H be the convex hull of A, B, $P_1, \cdots, P_n$; then the order (from A to B) of $P_1, \cdots, P_n$ is simply the order of the vertices on the boundary of H. [1, Theorem 4.1.] After the curve has been decomposed into convex segments, the sorting of points along a curve segment is then reduced to first identifying convex segments composing the segment, next sorting points on each convex segment, and finally concatenating the sorted lists.

To sort m points along (some segment of) an algebraic curve of degree n, the algorithm presented in Ref. 1 costs $O(mn^3\beta[n]+m\log m)$ time, where $O(\beta[n])$ is the cost of finding the real roots of an algebraic equation of order n. The $O(mn^3\beta[n])$ term covers the cost of locating the points on convex segments.

In this paper we use the well-known Cylindrical Algebraic Decomposition (CAD) [2, 3] of plane curves to solve the sorting problem. Cad decomposes an algebraic curve into monotone segments such that distinct points on such a segment correspond to distinct abscissae, and can thus be sorted by their abscissae. Such a monotone segment will be called an edge for reasons that will become clear later. In our approach, locating a point on an edge costs only $O(n^2+\alpha[n])$ time, where $\alpha[n]$ is the cost of computing the number of real roots of an order n algebraic equation within a certain interval. The total time complexity for sorting m points turns out to be $O(mn^2+m\alpha[n]+m\log m)$. We will also see that both constructing the Cad and constructing the convex decomposition have basically the same complexity, and will thus establish that the Cad method of sorting is superior to the convex decomposition method.

Another objective of this paper is to clarify some facts concerning the geometric structure of real

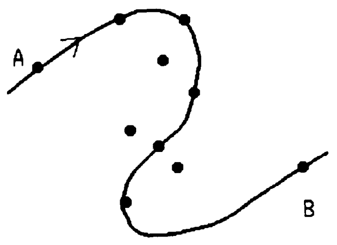

Fig. 1 Sorting points along curve segment AB.

*School of Engineering, Nagoya University, Furo-cho, Chikusaku, Nagoya, Japan.

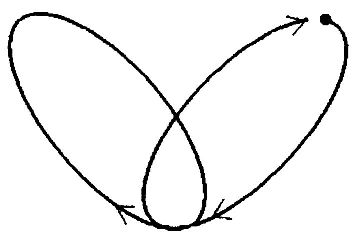
Fig. 2 Tracing along curve $2x^4-3x^2y+y^2-2y^3+y^4=0$.

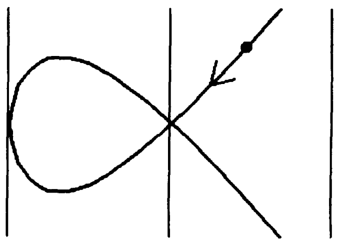
Fig. 3 Tracing along curve $(1-x)y^2-x^2(1+x)=0$.

algebraic curves. Recall that the order of points on a curve segment has been defined as the order in which the points would be encountered in traveling from one end point to the other along the segment. This definition depends essentially on a tangential direction continuously defined on the curve. However, at some singular points, tangential direction is not sufficient to define a unique tracing path. This problem has been settled many times in the literature [4, 1, 5], always by the use of quadratic transformations [6] to resolve the singular point. Here we would like to point out explicitly that any such transformation would determine the same tracing path.

A natural question is whether we could define a total order on the whole curve. To answer this question, suppose that we start tracing a curve from a smooth point, in a tangential direction, and continue tracing until the stage at which we are about to retrace some segment of the curve. This tracing will result in either a closed cycle (possibly with self crossing), as shown in Fig. 2, or a chain unbounded on one side (called a half chain in the sequel), as shown in Fig. 3. In the latter case, if we trace in the opposite direction from the starting point of the half chain, we obtain another half chain, which adds up with the original one to a chain unbounded on both sides (later referred to simply as a "chain"). In general, an algebraic curve may have several such cycles and/or chains, which uniquely determine a global order on the curve (We avoid saying "total order" for obvious reasons.).

It is clear that chains are nothing but open affine parts of cycles in the projective plane. for this reason, we will refer to the decomposition of a curve into cycles and chains as the cyclic structure of the curve. Proof has been given for a theorem [7] that for a projective curve of genus g, there are at most $g+1$ cycles. From this we can deduce that for an affine plane curve of degree n and genus g, the total number of cycles and chains is at most $g+n$. The genus of a plane curve of degree n satisfies: $g \leqq 1/2(n-1)(n-2)-1/2\Sigma\, r_P(r_P-1)$, where r_P is the multiplicity of point P, and P runs over all the singular points [6]. The equality holds if all the singular points (if any) are ordinary.
[Examples] The quartic

$$2x^4-3x^2y+y^2-2y^3+y^4=0$$

shown in Fig. 2 with two double poins has genus $\leqq 1$ (in fact, 0), and the cubic

$$(1-x)y^2-x^3-x^2=0$$

shown in Fig. 3 with an ordinary double point has genus 0.

The next section first briefly reviews the Cad decomposition of plane algebraic curves and then shows how to locate points on edges (monotone segments) obtained in the Cad.

In Section 3, we show how to connect edges into cycles and chains. We also discuss the main difficulties in constructing the Cad and the cyclic structure of a curve, comparing them with the difficulties of convex decomposition.

In Section 4, we discuss efficient computational strategies for locating points with approximate coordinate data.

2. Cad and the Structure Graph of an Algebraic Curve

2.1 Cad and the Structure Graph of an Algebraic Curve

We will first briefly explain some necessary facts about the Cad of plane algebraic curves, and then show how to apply a graph naturally associated with the Cad, called the structure graph, to the point sorting problem.

Assume that we are given a curve defined by an algebraic equation $F(x, y)=0$, where the polynomial

$$F(x, y)=f_0(x)y^n+f_1(x)y^{n-1}+\cdots+f_n(x)$$

has no factor that is a polynomial in variable x only, and no multiple factor. Let $R(x)$ be the discriminant [6] of $F(x, y)$ with respect to variable y; then $R(x)$ has degree $\leqq n(n-1)$, and the zeros of $R(x)$ correspond to the abscissae of the singular points and points at which the tangents to the curve are vertical. Further, if the curve has vertical asymptotes, their abscissae must be the zeros of $f_0(x)$ and hence zeros of $R(x)$, since $f_0(x)$ divides $R(X)$. Thus the vertical lines drawn at the zeros of $R(x)$ will divide the plane into strips, each of which contains several smooth segments of the curve, so that every such segment is monotone with respect to the xaxis; that is, distinct points on the segment correspond to distinct abscissae.

As an example, we consider the curve

$$F(x, y)=(1-x)y^2-x^2(1+x)=0.$$

whose discriminant $R(x)$ is given by

$$R(x)=\text{resultant}(F_y(x, y), F(x, y))$$
$$=\begin{vmatrix} -x^2(1+x) & 0 & 1-x \\ 0 & 2(1-x) & 0 \\ 0 & 0 & 2(1-x) \end{vmatrix}$$
$$=-4x^2(1+x)(1-x)^2.$$

It has three zeros -1, 0 and 1, corresponding to a vertical tangent line, a singular point, and an asymptote, respectively, as shown in Fig. 4.

Suppose that $R(x)$ has r distinct zeros; the plane is then divided into $r+1$ strips (including the left and right half planes) by r vertical lines. We index the strips from left to right as $S_1, \cdots, S_{r+1}$, and the lines as $l_1, \cdots, l_r$. Since degree $R(x)\leqq n(n-1)$, $r=O(n^2)$. In Fig. 4, the plane is divided into four strips by three vertical lines.

If the curve has p_i points on the line l_i, we define them as vertices $V_i^1, \cdots, V_i^{p_i}$, sorted from below. If the curve has q_i segments in the strip S_i, we define them as edges $E_i^1, \cdots, E_i^{q_i}$, sorted from below. These vertices and edges naturally form a graph. This graph may slightly differ from the usual graph in that it may have an edge with only one end point (vertex) or without an end point (In this case we may consider that the edge has void endpoints.).

Such a graph was called a Structure graph (S-graph) for the curve in [8]. We will at times adopt this terminology in this papere.

The data of incidence of edges with vertices of this graph can be computed be using an existing algorithm [2 (II), section 3]. An alternative method, which computes this data, as well as some other essential data, will be outlined in Section 3. But for the moment, we do not need this data in order to locate points on edges.

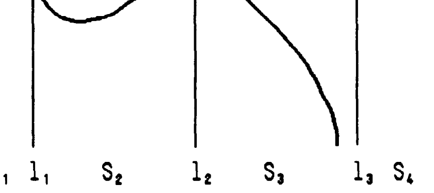

Fig. 4 Cad of curve $(1-x)y^2-x^2(1+x)=0$.

2.2 Locating Points on the Structure Graph

Let P be a point of the curve, with coordinates (a, b). We first determine the strip (or possibly some vertical line) in which P lies by comparing the abscissae of P with the vertical lines. Since the number of vertical lines $r=O(n^2)$, we need $O(\log(n^2))=O(\log n)$ comparisons. Next, if P lies on some l_i, comparing the ordinates identifies P with a vertex. On the other hand, if P lies in strip S_i, then the number of roots in the interval $(-\infty, b]$, of equation $F(a, y)=0$, is the second index j of the edge E_i^j on which P lies. This number can be calculated by using Sturm's theorem. Here we state a version of the theorem, the proof of which can be obtained by slightly modifying that of the conventional version of the theorem [e.g. 9, p. 244].

Sturm's theorem: Let $f(x)$ be a polynomial with real coefficients, without multiple zeros. Let $f_0, \cdots, f_k$ be a sequence of polynomials constructed as follows:

$$f_0=f, f_1=f',$$
$$f_{i-1}=f_i g_i-f_{i+1}$$

with $\deg f_{i+1}<\deg g_i$, for $i=1, \cdots, k-1$, where f_k is a nonzero number. For a real number b, let $v(b)$ be the number of changes of the signs in the sequence

$$f_0(b), f_1(b), \cdots, f_k(b) \text{ (neglecting zeros).}$$

The number of zeros of f in the interval $(b, +\infty)$ is then equal to $v(b)$.

In our application, we compute the number v of zeros of $F(a, y)$ in the interval $(-\infty, b)$, and obtain the index $j=v+1$.

The sequence $f_0, \cdots, f_k$ can be constructed by using Euclid's algorithm, which costs $O(n^2)$ time.

Since $f_i(x)$ in the above sequence has degree $\leqq n-i$, evaluating $f_i(b)$ takes $O(n-i)$ time. Therefore, for computing the index j, the Sturm's theorem provides an algorithm that costs $O(n)+O(n-1)+\cdots O(1)=O(n^2)$ time. Of course, we may use any other efficient method to compute the number of real roots. Throughout this paper, we simply assume that this computation need $O(\alpha[n])$ time.

It is not difficult to see that calculating the coefficients of $F(a, y)$ takes $O(n^2)$ time. Hence we conclude that locating a point takes $O(\log n)+O(n^2)+O(\alpha[n]) =O(n^2+\alpha[n])$ time.

3. Cycles and Chains on an Algebraic Curve, and Sorting Points along an Algebraic Curve

3.1 Connecting Edges into Cycles and Chains

In the Inbroduction we explored the cyclic structure of an algebrain curve. Here we will see how each cycle and chain is constructed from the edges of the S-graph of the curve.

It is clear that every edge of the S-graph must be entirely contained in a single cycle or chain, and conversely, that each cycle and chain must be composed of several edges (or possibly just one edge). Therefore we can represent it by a list of edges sorted in cyclic or linear order, and represent the whole curve by a set of such lists. Note that in this representation, the cycles are special kinds of graph theoretic cycles of the structure graph.

[Examples] As shown in Fig. 5(a), the curve

$$(1-x)y^2-x^3-x^2=0$$

has one chain, represented by the list of edges [E_3^2, E_2^1, E_3^2, E_3^1].

In Fig. 5(b), the curve

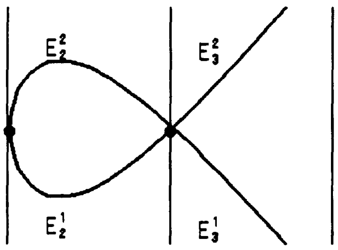

Fig. 5(a) Curve $(1-x)y^2-x^2(1+x)=0$ has a chain [E_3^2, E_2^1, E_2^2, E_3^1].

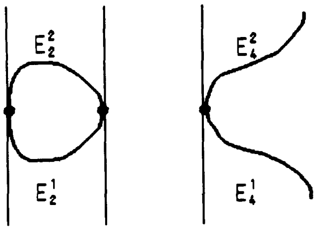

Fig. 5(b) Curve $y^2-x^3+x=0$ has a cycle [E_2^1, E_2^2] and a chain [E_4^2, E_4^1].

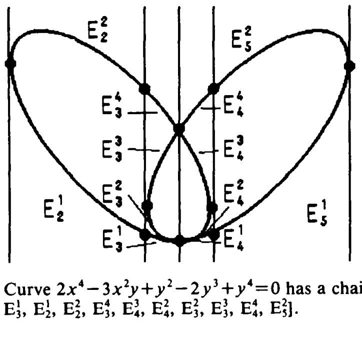

Fig. 5(c) Curve $2x^4-3x^2y+y^2-2y^3+y^4=0$ has a chain [E_5^1, E_4^1, E_3^1, E_2^1, E_2^2, E_3^4, E_4^3, E_4^2, E_3^2, E_3^3, E_4^4, E_5^2].

$$y^2-x^3+x=0$$

has a cycle [E_2^1, E_2^2] and a chain [E_4^2, E_4^1].

In Fig. 5(c), the curve

$$2x^4-3x^2y+y^2-2y^3+y^4=0$$

has one cycle [E_5^1, E_4^1, E_3^1, E_2^1, E_2^2, E_3^4, E_4^3, E_4^2, E_3^2, E_3^3, E_4^4, E_5^2].

With the notations r, p_i, l_i, V_i^j, S_i, S_{i+1} as in the previous section, the following flowchart (Fig. 6) shows the steps for connecting edges into cycles and chains.

The following will be computational strategies for implementing steps ①, ②, ③ and ④ in the flowchart.

① Computing the Asymptotic Edges.

For every vertical line l_i in the Cad (cf. Section 2), we first compute how many edges in its left strip and right strip, respectively, are asymptotic to the lower half of l_i. The reason for this will become clear in the sequel [2, II].

We choose a real number b smaller than the ordinate of every vertex of the S-graph. Suppose that the line $y=b$ intersects the curve at points (a_1, b), $\cdots$, (a_k, b), which are sorted so that $a_1<\cdots<a_k$. (Note that $k\leq n$, the degree of the curve) (Fig. 7). Let a_{ki} be the largest root that is smaller than x_i, the abscissa of l_i. Choose a real number c such that $x_{i-1}<c<x_i$ and $a_{ki}<c$. Now the number of intersections of the curve with the halfline drawn downward from point (c, b) will be the number of edges in S_i that are asymptotic to the lower half of l_i. Obviously, similar methods can be applied to the strip S_{i+1}.

Note that the data (a_1, b), $\cdots$, (a_k, b) are fixed for all the vertical lines.

Suppose that there are s edges in S_i and t edges in S_{i+1} asymptotic to the lower half of l_i, computed as above. We next show how to compute the data of connectivity between edges (in S_i and S_{i+1}) that are incident to vertex V_i^1, the first vertex from below on l_i.

② Computing the Data of Connectivity at a Smooth Vertex.

If V_i^1 is a smooth point, then there will be three cases, as shown in Fig. 8. These can be distinguished as follows:

Let P be a point below V_i^1 on l_i, and Q a point between V_i^1 and V_i^2.

If the evaluations of $F(x, y)$ at the two points have different signs, that is, if $F(P)F(Q)<0$, then edge E_i^{s+1} is connected with E_{i+1}^{t+1} at vertex V_i^1, as shown in Fig. 8(a).

If $F(P)F(Q)>0$, assume that $F(P)$ (and hence also $F(Q)>0$ In this case, if the normal vector (F_x, F_y) at V_i^1 points to the right, that is, if $F_x>0$, as shown in Fig. 8(b), then the two edges connected at V_i^1 are E_i^{s+1} and E_i^{s+2}; otherwise, the two edges will be E_{i+1}^{t+1} and E_{i+1}^{t+2}, as shown in Fig. 8(c). A similar argument can be given for $F(P)<0$. We conclude that condition $F(P)F_x(V_i^1)>0$ completely specifies case (b) and that $F(P)F_x(V_i^1)<0$ specifies case (c).

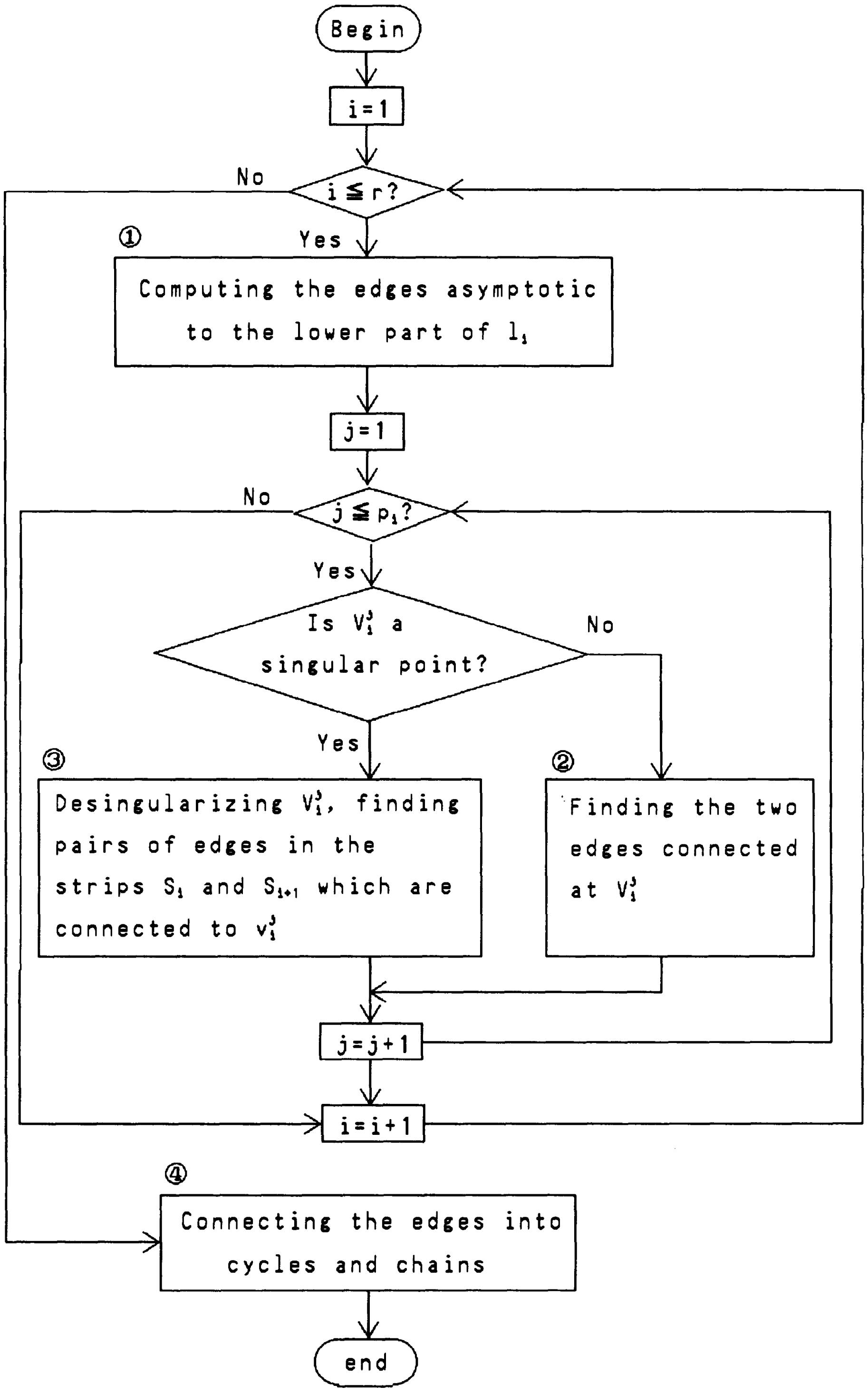
Begin
i=1
i ≦ r?
No
Yes
①
Computing the edges asymptotic to the lower part of l_i
j=1
j ≦ p_i?
No
Yes
Is V_i^j a singular point?
No
Yes
③
Desingularizing V_i^j, finding pairs of edges in the strips S_i and S_{i+1} which are connected to v_i^j
②
Finding the two edges connected at V_i^j
j=j+1
i=i+1
④
Connecting the edges into cycles and chains
end

Fig. 6 Flowchart for connecting the edges into cycles and chains.

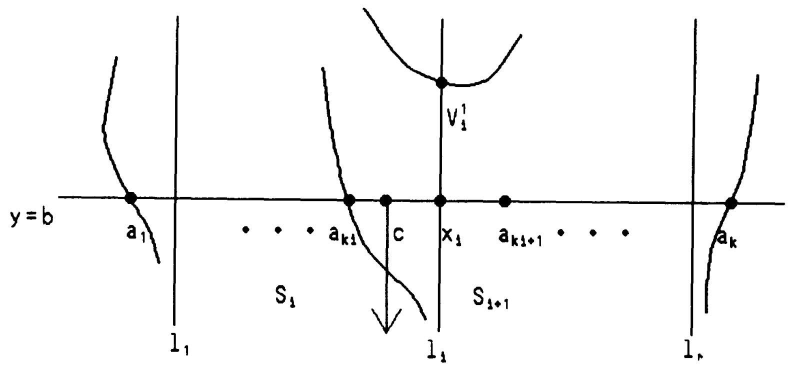

Fig. 7 Line $y=b$, drawn below all vertices, intersects with the curve at abscissae $a_1, \cdots, a_k$.

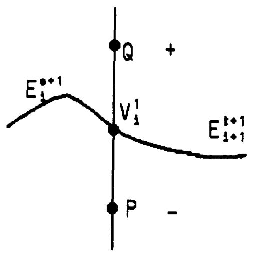

Fig. 8(a) The case $F(P)F(Q)<0$.

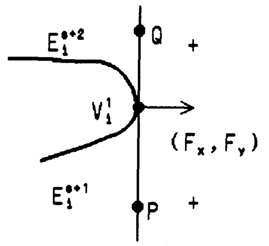

Fig. 8(b) The case $F(P)F(Q)>0$ and $F(P)F_x(V_i^1)>0$.

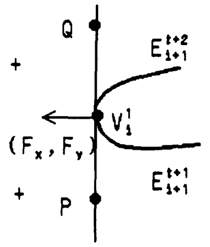

Fig. 8(c) The case $F(P)F(Q)>0$ and $F(P)F_x(V_i^1)<0$.

③ Computing Pairs of Edges Connected at a Singular Vertex.

If V_i^1 is a singular point, we will use quadratic transformations to identify and pair up edges incident to V_i^1. Since this subject has been intensively discussed in the literature [1, 4], many details will be omitted in the following description.

Since an edge can be uniquely represented by a point on it, the procedure for locating the point, as described in section 2, would identify the edge; thus our goal will be to obtain a set of pairs of points, such that every two edges that should be paired up are represented by a pair of points in the set.

Suppose that after a sequence of quadratic transformations, we obtain a smooth point P (in the real plane), which is the transform of the original singular point. we choose a suitable pair of points near P on the transformed curve, and reverse the transformations. We are then expected to obtain a pair of points on two edges incident to V_i^1, which are to be paired. In this way, we can obviously reach our goal.

[Example] As shown in Fig. 9, smooth point P is a transform of V, and two points A'' and B'' near P are transformed back into points A and B, which specify two edges (E_3^1 and E_4^1 in Fig. 5(c)) to be connected at V.

Remark: An alternative computational strategy for connecting segments at a singular point would be to employ a formal power series expansion. A real place [6] centered at the singular point corresponds to a pair of edges to be connected.

One problem is that we do not know any criterion for deciding from the first several terms of the expansion. Whether a place should be real and whether one real place can be distinguished from another.

So far, we have shown how to identify edges incient to vertex V_i^1, and how to pair the edges if V_i^1 is a singular point. If we deploy a similar procedure successively to vertices $V_i^2, \cdots, V_i^{p_i}$ on line l_i, we obtain the data on the incidence of the edges to these vertices and on the connectivity between edges.

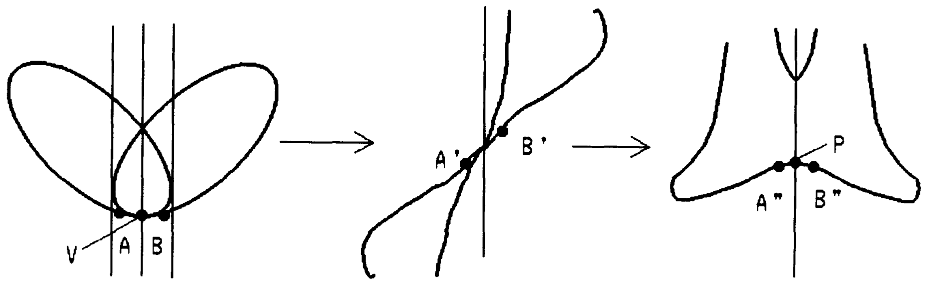

Fig. 9 Pairing up segments by desingularization.

④ Connecting Edges into Cycles and Chains.

With these data, it will be easy to connect edges into cycles and chains. In Section 1, we described how to trace a curve to obtain cycles and chains provided the segments are well paired up at singularities. The graphical version of this tracing obtains cycles, which are represented as lists of edges according to the connectivity data computed as above. We may also define the direction of an edge so that the first traced vertex is the head and the next traced vertex the tail of the edge; this definition will make it easy to identify an arbitrary segment with edges, as we will describe later.

We would first like to say a few words about what a "curve segment" or simply a "segment" means. Corresponding to the cyclic structure of an algebraic curve we constructed, it seems improper that a segment should run through several distinct cycles and chains. Therefore we consider it a good convention to require that a segment be contained in a single cycle or chain, as is probably assumed implicitly in all the literature.

Since we have put a direction on every edge in a general cycle or chain, this gives the cycle a global orientation. Therefore, an ordered pair of points uniquely defines a segment as the edges and/or subedges (parts of edges) of a cycle or chain traced in that orientation from the first point to the second point.

3.2 Algorithms for Sorting Points along an Algebraic Curve

It is now trivial to give an algorithm for sorting points along a curve segment. Suppose we are given m points. We first locate them on the edges or subedges of the segment. Next, we sort points on each edge according to the direction of the edge (for example, if the head of an edge has a smaller abscissa than the tail, then the points on it should be sorted in ascending order of their abscissae). Finally, concatenating the sorted lists of points, we obtain the whole list of points sorted in the order in which they are encountered in traveling along the segment.

If we want to sort points along the entire curve, we can obtain several sorted lists of points, each of which represents the points sorted on some cycle or chain.

As we described in Section 2, locating m points takes $O(mn^2+m\alpha[n])$ time, where $\alpha[n]$ is the cost of calculating the number of points of a half line with a curve of degree n.

Let m_i be the number of points on some edge that are to be sorted according to their abscissae. With any optimal sorting algorithm, this sorting costs $O(m_i \log m_i)$ time. The sum of the time over all the individual edges is then $O(\Sigma(m_i \log m_i)) \leqq O((\Sigma m_i)\log(\Sigma m_i)) = O(m \log m)$. Concatenating the sorted lists costs $O(m)$ time, so the total time for sorting and concatenating is $O(m \log m) + O(\mathrm{m}) = O(m \log m)$.

Therefore, the total time complexity for sorting m points along the curve is $O(mn^2 + \alpha[n] + m \log m)$.

To compare this complexity with the complexity $O(mm^3\beta[n\} + m \log m)$ of the convex decomposition approach of Jonhstone and Bajaj, we need to compare $\alpha[n]$ with $\beta[n]$, the complexity of computing all the real roots of the order n equation. For implementation, if one uses the Sturm's method both to compute the number of real roots and to compute the real roots themselves, then obviously $\alpha[n] \leqq \beta[n]$. Even if the roots are computed by practically efficient methods such as the Durand-Kerner method, one may count the number of roots in an interval in linear time, and we still have $O(\alpha[n]) = O(\beta[n])$. Therefore we conclude that for sorting points along an algebraic curve the Cad method is much more efficient than the convex decomposition method.

3.3 Comparison of Complexities for Constructing the Cad and the Convex Decomposition

We will not try to formulate a bound for the time complexity of constructing the Cad and cyclic structure of an algebraic curve, but will content ourselves with giving some evidence to argue that the Cad approach could not be more complicated than the convex decomposition approach.

First, we note that the pairing up of segments at singular points has the same complexity for both the Cad approach and the convex approach, as long as quadratic transformations are adopted in both.

Secondly, while the Cad decomposition of a degree n curve solves an order $n(n-1)$ polynomial equation (the discriminant) to get the vertical lines that divide the

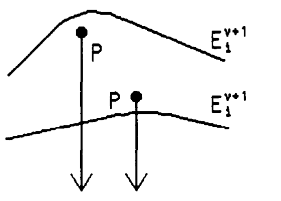

Fig. 10 Is P approximately on E_i^v or E_i^{v+1}?

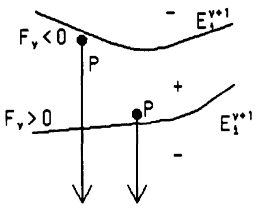

Fig. 11(a) The case $F(P)>0$.

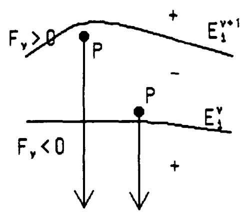

Fig. 11(b) The case $F(P)<0$.

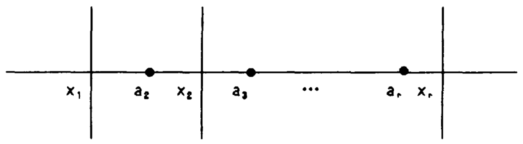

Fig. 12 a_i is a rational number in the interval (x_{i-1}, x_i), for $i=2, \cdots, r$.

plane into strips, the convex decomposition must solve systems of polynomial equations to find the flexes and singularities of the curve [1, Section 5.5]. (In the Cad approach, singularities are vertices of the S-graph, and can be distinguished from smooth points simply by checking whether $f_x=f_y=0$.)

Thirdly, while the Cad computes the intersections of the curve with $O(n^2)$ vertical lines, the convex decomposition computes the intersections of the curve with $O(n^2)$ tangent lines of flexes and singularities.

It is clear that the computations addressed above are the main difficulties in the two approaches. We therefore conclude that the construction of the Cad and cyclic structure of an algebraic curve will not be more expensive than convex decomposition.

4. Some Numerical Considerations Regarding Implementation

In this section, on the assumptions that polynomial $F(x, y)$ has rational coefficients, and that the Cad and cyclic structure of the curve $F(x, y)=0$ are correctly constructed (with symbolic computation), we discuss techniques for sorting points with approximate coordinate data.

For a general algebraic curve, we can expect only a finite number of rational points (points with rational coordinates) on it (Mordell's conjecture [10, p. 266]). This implies that if we are given a set of points with rational coordinates, it is better to assume that the points are only approximately on the curve. Approximate data can raise serious violations during locating the points on the structure graph.

4.1 On the Use of Sturm's Theorem

The most disturbing problem arises in the use of Sturm's theorem. For locating a point $P=(a, \mathrm{b})$, Sturm's theorem computes the number v of roots of $F(a, y)=0$ in the open interval $(-\infty, b)$. Suppose P lies on edge E_i^j. Here j is taken to be equal to $v+1$. Since it is assumed that $F(x, y)$ has rational coefficients and that $P=(a, b)$ is rational, it is easy to calculate the correct v. If P is precisely on the curve, that is, if $F(a, b)=0$, then we have $j=1+v$. On the other hand, if $F(a, b)\neq 0$, should we assume that $j=1+v$ or $j=v$? Both cases seem equally likely to happen (Fig. 10).

An observation that may settle this problem is as follows: as shown in Fig. 11(a), if $F(a, b)>0$, then we must have $F_y>0$ near E_i^v and $F_y<0$ near E_i^{v+1}. Otherwise if $F(a, b)<0$, as shown in Fig. 11(b), we have $F_y<0$ near E_i^v and $F_y>0$ near E_i^{v+1}. This observation leads us to conclude the following simple criterion:

$$\begin{aligned} &j=v, \text{ if } F(a, b)F_y(a, b)>0; \\ &j=v+1, \text{ if } F(a, b)F_y(a, b)\leq 0. \end{aligned} \tag{1}$$

4.2 A Technique for Comparison of Abscissae

Another problem that we will discuss arises in the comparison of the abscissa of P with the abscissae of vertical lines when determining the strip S_i in which P lies.

Because of the construction of the Cad of the curve, the abscissae of the vertical lines are not rational in general. If we want to calculate the index i correctly,we certainly cannot approximate these abscissae with rational numbers. A scheme that involves only rational numbers and still computes the correct index is as follows:

As in Section 2, let the abscissae of the vertical lines, that is, the real roots of the discriminant $R(x)$ of $F(x, y)$, be $x_1<\cdots<x_r$. In the construction of the Cad, we may want to obtain a sequence of rational numbers $a_2, \cdots, a_r$ such that $x_1<a_2<x_2<\cdots a_r<x_r$ (Fig. 12).

Since $F(x, y)$ is assumed to have rational coefficients, $R(x)$ also has rational coefficients, namely $R(x)$ $Q[x]$. Suppose $R(x)=(A_1(x))^{e_1}\cdots(A_r(x))^{e_r}$, where each $A_i(x)$ $Q[x]$, $i=1, \cdots, r$, is irreducible in $Q[x]$. Let $R_1(x)=A_1(x)\cdots A_r(x)$. $R_1(x)$ then has no multiple factor in $Q[x]$, and hence no multiple factor in $R[x]$ (see Chapter I, Theorem 9.5, [6]). Therefore $x_1, \cdots, x_r$ are simple roots of $R_1(x)$, and $R_1(x)$ takes the values of opposite signs on any two adjacent intervals, such as $R_1(a_2)R_1(a_3)<0$.

Under these conditions, the strip S_i that contains P is determined as follows: We compare a with $a_1 = -\infty$, $a_2, \cdots, a_r, a_{r+1} = +\infty$. Suppose $a_k < a < a_{k+1}$ $(1 \leqq k \leqq r)$. Then we decide i as:

$$i = k, \text{ if } R_1(a) R_1(a_k) > 0; \qquad i = k+1, \text{ if } R_1(a) R_1(a_k) < 0. \tag{2}$$

5. Conclusions

We have developed algorithms for the problem of sorting points along an algebraic curve using the Cad of the curve. Our approach has less computational complexity than the convex deomposition approach of Johnstone and Bajaj, which was asserted to be superior to many conventional methods of sorting [1]. We have also explored the cyclic structure of real algebraic curves.

To our knowledge, the problem of sorting points along an algebraic curve was stated clearly and treated systematically for the first time in [1] in which many interesting related problems were also discussed. We think that these kinds of problem are most significant in that they motivate the study of the geometric structure of general algebraic objects, which ae important in developing advanced geometric modeling techniques. In this paper the structure of real algebraic curves has been made rather clear. Naturally, our next ambition is to do similar work on surfaces.

Acknowledgements

We thank Professor N. Onishi and Miss N. Matsuzaki of Nagoya University for valuable discussions and warm help. We are also grateful to the reviewers, whose comments greatly improved this paper.

References

1. Johnstone, J. K. and Bajaj, C. L. Sorting points along an algebraic curve, *SIAM J. Comput.* **19**, 5 (1990), 925-967.
2. Arnon, D. S., Collins, G. E. and MacAllum, S. Cylindrical algebraic decomposition, I, II, *SIAM J. Comput.* **13** (1984), 865-889.
3. Collins, G. E. Quantifier elimination for real closed fields by cylindrical algebraic decomposition, *in Springer Lecture Notes in Comp. Sci.*, **33**, Springer-Verlag, Berlin (1975).
4. Bajaj, C., Hoffmann, C., Lynch, B. and Hopcroft, J. Tracing surface intersections, *Comput. Aided Geom. Des.*, **5** (1988), 285-307.
5. Hoffmann, C. M. Geometric & Solid Modeling, an Introduction, Morgan Kaufmann Publishers, Inc., San Mateo, California (1989).
6. Walker, R. Algebraic Curves, Princeton Univ. Press, N.J. (1950).
7. Ying, J. Q. Computations on real algebraic curves, Ph.D. thesis, Department of Electrical Engineering, Nagoya University, Nagoya, Japan (1992).
8. Arnon, D. S. Topologically reliable display of algebraic curves, *Computer Graphics*, **17**, 3 (1983), 219-227.
9. Waerden, V. D. Algebra, 7-th Edition, Springer-Verlag (1991).
10. Silverman, J. H. The Arithmetic of elliptic curves, Graduate Text in Mathematics, Springer-Verlag (1986).

(Received August 12, 1991; revised February 28, 1992)