TRACING SURFACE INTERSECTIONS

C. L. Bajaj
C. M. Hoffmann
J. E. Hopcroft
R. E. Lynch

CSD-TR-728
December 1987

This report supercedes CSD-TR 637 and CSD-TR 684.

Tracing Surface Intersections

C. L. Bajaj[*]
C. M. Hoffmann[†]
J. E. Hopcroft[‡]
R. E. Lynch[§]

Abstract

We consider the problem of tracing the intersection of surfaces given either implicitly or parametrically. We give a numerical tracing procedure in which a third-order Taylor approximant is constructed for taking steps of variable length, and the points so found are improved by Newton iteration. We show how this construction relates to local parameterizations of the curve at singularities, and discuss our experience with the method. For plane curves, given implicitly, we show how desingularization techniques can be incorporated to trace correctly through all types of singularities. An implementation of this method is also discussed.

1 Introduction

A basic operation recurring in geometric modeling is the evaluation of space curves given as the intersection of two surfaces. Existing geometric modeling systems typically restrict the geometric coverage, that is, the allowed faces may be planar [26], natural quadrics [21], arbitrary quadrics [14, 18], or parametric patches of various types [6, 17]. With such specializations

[*]Computer Science Department, Purdue University. Supported in part by NSF Grant MIP 85-21356

[†]Computer Science Department, Purdue University. Supported in part by NSF Grants DCR 85-12443 and CCR 86-19817 and ONR contract N00014-86-K-0465

[‡]Computer Science Department, Cornell Univerity. Supported in part by NSF Grants DCR 85-02568 and DMC 86-17355 and ONR contract N00014-86-K-0281

[§]Computer Science Department, Purdue University.

many good techniques can be developed that take advantage of the specific restrictions.

In this paper, we consider the evaluation of surface intersections in general. The intersecting surfaces may specified implicitly as $f_1(x, y, z) = 0$ and $f_2(x, y, z) = 0$, where f_1 and f_2 are smooth functions, or parametrically as $(x = G_{1,1}(u_1, v_1), y = G_{2,1}(u_1, v_1), z = G_{3,1}(u_1, v_1))$ and $(x = G_{1,2}(u_2, v_2), y = G_{2,2}(u_2, v_2), z = G_{3,2}(u_2, v_2))$, where the $G_{i,j}$, $i = 1, 2, 3$, $j = 1, 2$, are smooth functions. In full generality, tracing the intersection curve is a difficult problem, and one of our objectives is to explore the scope of a purely numerical approach. [20] reviews several such methods. A common problem stems from the inherent geometric complexity of high degree algebraic curves that arise as curves of intersection. In particular, such a curve may possess singular points where the curve has an abrupt change of normal direction (cusps), multiple self-intersecting branches (nodes), or self-tangent branches (tacnodes). In the neighborhood of a singularity the determinant of the linear system used to approximate the curve locally approaches zero. Purely numerical tracing schemes have great difficulties in this situation: As the singularity is approached, these programs may fail. Even if they trace through the singularity without mishap, they may identify the curve branches incorrectly.

It is not known how to rectify all these difficulties with a single numerical method. Nevertheless, it is our experience that a carefully crafted numerical tracing routine is capable of handling many of the difficulties characterized above. We propose here such a scheme in which the intersection curve is locally approximated by a low degree Taylor polynomial interpolant, and a new curve point estimate is derived from it by taking steps of variable lengths. Newton iteration is then used to refine this new point estimate.

A strength of the method lies in its ability to consolidate the computation needed for the Newton iteration with the computation determining the power series expansion. Moreover, as we show, there is a close correspondence of the computational machinery needed by the method with an algebraic procedure for analyzing the curve at singular points. Although this correspondence is not exploited in this paper, it permits a fairly simple extension to cope directly with a large class of singularities. Another advantage of the approach is that we can construct higher order approximants to the intersection of parametric surfaces directly. Previously, only piecewise linear techniques have been used that are constructed either from subdivision methods [7], or directly from the equations. In the latter case, a step length constraint is added to avoid solving an underdetermined system [20]. However, as we have found, there is no difficulty in solving the underconstrained system and the step length constraint is artificial.

Next, we consider the special case of tracing plane algebraic curves defined implicitly as $f(x,y) = 0$. Tracing plane curves which are given parametrically simply amounts to evaluating the parametric equations for several distinct parameter values. So, one could try to obtain a rational parameterization of f. Only curves of genus zero possess a rational parametric form, however. For algorithms to test whether and how implicitly defined plane curves can be rationally parameterized, see [4].

The tracing of implicitly defined plane curves arises in solid modeling in a number of ways:

1. When the faces of a model are parametric patches, with known implicit equations, edges bounding these patches can be represented as plane curves in the parameter plane of one of the faces, see [9].

2. When intersecting two implicit surfaces $f_1(x,y,z) = 0$ and $f_2(x,y,z) = 0$, one of them, say f_1, may possess a rational parameterization. By substituting the parametric equations of f_1 into the implicit equation of f_2, a plane curve in the parameter plane is obtained that is in birational correspondence with the intersection curve of f_1 and f_2. For efficient algorithms to test whether and how an implicit quadric or cubic algebraic surface can be parameterized, see [2, 3]

3. When intersecting nonrational implicit surfaces $f_1(x,y,z) = 0$ and $f_2(x,y,z) = 0$, one can always find a rational surface $f_3(x,y,z) = 0$ containing the intersection curve of f_1 and f_2. After f_3 has been found, it is easily parameterized, and we can obtain a plane curve by substituting as in 2 above. For methods to achieve this for irreducible algebraic space curves see [5,12,22].

Here, birational correspondence means that in each direction rational maps exist. In consequence, a tracing procedure for plane algebraic curves yields a tracing procedure for algebraic space curves. Note, however, that the corresponding plane curve might have more singularities than the space curve. Moreover, the degree of the curve is the product of the surface degrees, so that tracing the corresponding planar curve is numerically more delicate. If the birational map is not derived carefully, finally, the degree of the plane curve may be even higher. Thus, for simple singularities, the purely numerical approach remains attractive.

We show that for plane algebraic curves the correct branch connectivity can be achieved by utilizing results from algebraic geometry. The trace of $f(x, y) = 0$ commences at a given input point with a desired direction. At noncritical segments, we proceed numerically as before. When the condition number of the system becomes very large, we try to locate a nearby curve singularity. Then, by applying quadratic transformations, the branch of f we trace is birationally mapped to a branch of a transformed curve g that has no singularities. The transformed branch is traced and the points of g are mapped to corresponding points of f. The trace of g continues until we have passed the singularity of f. In this way, correct branch connectivity is achieved.

2 Notation and Definitions

Partial derivatives are written by subscripting, for example, $f_x = \partial f/\partial x$, $f_{xy} = \partial^2 f/(\partial x \partial y)$, and so on. Since we consider analytic curves and surfaces, we have $f_{xy} = f_{yx}$ etc.

Vectors and vector functions are denoted by bold letters. The *inner product* of vectors $\mathbf{a}$ and $\mathbf{b}$ is denoted $\mathbf{a} \cdot \mathbf{b}$. The *length* of the vector $\mathbf{a}$ is $\|\mathbf{a}\| = \sqrt{\mathbf{a} \cdot \mathbf{a}}$.

The *gradient* of f is the vector $\nabla f = (f_x, f_y, f_z)$. The *Hessian* of f is the symmetric matrix

$$H_f = \begin{pmatrix} f_{xx} & f_{xy} & f_{xz} \\ f_{yx} & f_{yy} & f_{yz} \\ f_{zx} & f_{zy} & f_{zz} \end{pmatrix}$$

The intersection of f_1 and f_2 is denoted by $\mathbf{r}(s)$ and is a vector function of the argument s, typically the arc length measured from some point on the curve. Derivatives of $\mathbf{r}(s)$ are denoted $\mathbf{r}'$, $\mathbf{r}''$, $\ldots$, $\mathbf{r}^{(m)}$.

A point $p = (x, y, z)$ is *regular* on f if the gradient of f at p is not null; otherwise the point is *singular*. A point p of the intersection $\mathbf{r}(s)$ is *regular* if p is regular on both f_1 and f_2 and if the gradients ∇f_1 and ∇f_2 are linearly independent. That is, the surfaces are not singular at p and intersect transversally.

If one of the surfaces is a plane, then a simple coordinate transformation reduces the problem to tracing a plane curve $f(x, y) = 0$. Assume that this curve contains the origin and is algebraic. Then the *order form* is the homogeneous polynomial $F(x, y)$ consisting of the terms of lowest degree in f. It contains information about the curve's behavior at the origin. If the order form is linear, then the curve has a *simple* point at the origin, i.e., the curve is not singular at the origin. If the order form is nonlinear, then the origin is a *singularity*. The degree of F is then called the *order* of the singularity. Moreover, the linear factors of F are equations of the *tangents* of the curve at the origin.

An important concept from algebraic geometry, used to study the local curve structure, is that of *place*, e.g., [25]. Briefly, a *place* of $f(x, y) = 0$ is a pair of power series

$$x(s) = a_0 + a_1 s + a_2 s^2 + \dots$$

$$y(s) = b_0 + b_1 s + b_2 s^2 + \dots$$

such that $f(x(s), y(s))$ is identically zero. The place is said to be *centered* at the point $(x(0), y(0))$ of the curve. It is always possible to choose the place such that $x(s) = s^k$, for some k. Intuitively, a place is a local parameterization of the curve, centered at $(x(0), y(0))$, with a certain radius of convergence that varies with the place.

If the center c is not a singular point, then the place is equivalent to the Taylor series about c. If the c is singular, then the curve may have more than one distinct place centered at c, each corresponding to a distinct branch of the curve.

The *order* of a place centered at the origin is the lowest exponent with a nonzero coefficient in the power series. For example, the order of

$$x(s) = s^2$$

$$y(s) = s^3$$

is two, whereas the order of

$$x(s) = s$$

$$y(s) = s + s^2/2 - s^3/8 + - \dots$$

is one.

Centered at every nonsingular point, the curve has exactly one linear place, i.e., a place of order one. At a singular point the curve has one or more places which may or may not be linear. However, if there is only one place at a singular point, then this place must be nonlinear.

3 Nonsingular Curve Points on Surface Intersections

We consider first tracing the intersection of implicit surfaces, $f_1(x,y,z) = 0$ and $f_2(x,y,z) = 0$, given an initial curve point and a direction. In the simplest situation we trace the intersection in a neighborhood in which both f_1 and f_2 are regular and their gradients are linearly independent. Geometrically this means that the surfaces intersect transversally and are not singular in the vicinity. We formulate a system of equations from which both the local approximation as well as the Newton iteration are derived. Under the assumption of linearly independent gradients, we have a system of linear equations of rank 2. The choices made when solving the system correspond to parameterizing the approximant by arc length.

We then sketch how this approach can be directly transferred to tracing the intersection of parametric surfaces, $(G_{1,1}(u_1,v_1), G_{2,1}(u_1,v_1), G_{3,1}(u_1,v_1))$ and $(G_{1,2}(u_2,v_2), G_{2,2}(u_2,v_2), G_{3,2}(u_2,v_2))$, with the $G_{i,j}$, $i = 1,2,3$, $j = 1,2$, as smooth functions, given an initial curve point and a desired direction. Again, higher order approximants are easily constructed and are useful for estimating a safe step length. Under the assumption of linearly independent gradients, we now have a system of linear equations of rank 3. It is clear that the approach generalizes to tracing the intersection of $n-1$ hypersurfaces in n-dimensional space.

3.1 Equation for the Intersection

We treat the case that the intersection $\mathbf{r}$ is a function, having at least four continuous derivatives, of a parameter s. Then

$$\mathbf{r}(s) = \mathbf{r}(0) + s\mathbf{r}'(0) + \frac{s^2}{2}\mathbf{r}''(0) + \frac{s^3}{6}\mathbf{r}'''(0) + \mathbf{e}(s) = \mathbf{p}(s) + \mathbf{e}(s), \quad (1)$$

where $\mathbf{p}$ is the cubic Taylor interpolant to $\mathbf{r}$ at $s = 0$ and $\mathbf{e}$ is its error, or remainder. Below we give a numerical procedure for finding values of the derivatives, given a point q_0 on the intersection. Since $e(s) = O(s^4)$ in a bounded interval containing $s = 0$, a sufficiently small s makes the value $p(s)$ of the cubic an accurate estimate of $r(s)$. Using $p(s)$ as an initial estimate, one can then obtain another point, q_1 on the intersection with a very few steps of Newton iteration. The process then repeats. In this way a sequence of points, q_n, $n = 0, 1, 2, \cdots$, on the intersection is determined.

The derivatives are necessarily not unique because the parameterization of r by s is nonunique. We choose s as arc length. Then the unit tangent t, the unit principle normal n, and the unit binormal b are related by the Frenet-Serret formulae [10, p. 107]:

$$\frac{dt}{ds} = \kappa n, \qquad \frac{db}{ds} = -Tn, \qquad \frac{dn}{ds} = Tb - \kappa t, \tag{2}$$

where $\kappa = 1/\rho$ is curvature and $T = 1/\tau$ is torsion. The vectors t, n, and b form an orthonormal triad with $n = b \times t$. With s arc length, the derivatives of r are given by

$$r'(s) = t, \qquad r''(s) = \frac{dt}{ds} = \kappa n,$$
$$r'''(s) = \left[\frac{d}{ds}(\kappa n) = \frac{d\kappa}{ds}n + \kappa\frac{dn}{ds} =\right] \kappa' n + \kappa T b - \kappa^2 t. \tag{3}$$

3.2 Implicit definition

First suppose that points on the curve are defined as solutions of $f_j(x, y, z) = f_j(r) = 0$, $j = 1, 2$. The Taylor expansion of $f_j(r(s))$ in powers of s is

$$\begin{aligned} f_j(r(s)) &= f_j(r(0)) + s\left[\frac{\partial f}{\partial x}\frac{dx}{ds} + \frac{\partial f}{\partial y}\frac{dy}{ds} + \frac{\partial f}{\partial z}\frac{dz}{ds}\right] + \cdots \\ &= f_j(r(0)) + s\nabla f_j \cdot r'(0) \\ &\quad + \frac{s^2}{2}[\nabla f_j \cdot r''(0) + r'(0) \cdot H_{f_j} \cdot r'(0)] + \cdots, \quad j = 1, 2, \end{aligned} \tag{4}$$

where ∇f_j is the gradient of f_j and H_{f_j} its Hessian, both evaluated at $r(0)$.

Since the intersection satisfies $f_j(r(s)) \equiv 0$, the coefficient of each power of s in (4) must be zero. Given a point $q = r(0)$ on the intersection, the m-th derivative of r then satisfies

$$\nabla f_j(q) \cdot r^{(m)}(0) = b_{j,m}, \quad j = 1, 2. \tag{5}$$

The quantities $b_{j,m}$ are expressed in terms of the partial derivatives of f_j and lower-order derivatives of $\mathbf{r}$; e.g.:

$$b_{j,1} = 0, \qquad b_{j,2} = -\mathbf{r}'(0) \cdot H_{f_j} \cdot \mathbf{r}'(0), \quad j = 1, 2;$$

for $b_{j,3}$, see Appendix **A.1**. For each m, (5) is a pair of equations for the three components of $\mathbf{r}^{(m)}(0)$. Appendix **A.2** details how to solve this system with numerically stable techniques.

It follows from the independence of the gradients that there is a unit vector $\mathbf{t}$ which is perpendicular to both gradients:

$$\nabla f_1 \cdot \mathbf{t} = \nabla f_2 \cdot \mathbf{t} = 0, \qquad \mathbf{t} \cdot \mathbf{t} = 1.$$

Except for sign, $\mathbf{t}$ is unique. Any vector can be written as a linear combination of these three; in particular

$$\mathbf{r}^{(m)} = \alpha_m \mathbf{t} + \beta_m \nabla f_1 + \gamma_m \nabla f_2.$$

Direct substitution into (5) yields

$$\beta_m \nabla f_j \cdot \nabla f_1 + \gamma_m \nabla f_j \cdot \nabla f_2 = b_{j,m}, \quad j = 1, 2. \tag{6}$$

There is a unique solution, β_m, γ_m, of this system and, therefore,

$$\mathbf{r}^{(m)} = \alpha_m \mathbf{t} + \beta_m \nabla f_1 + \gamma_m \nabla f_2, \tag{7}$$

with α_m arbitrary, is the general solution of the system (5).

Because $b_{1,1} = b_{2,1} = 0$ makes $\beta_1 = \gamma_1 = 0$, we have $\mathbf{r}'(0) = \alpha_m \mathbf{t}$. The choice $\alpha_m = 1$ makes $\mathbf{r}'(0)$ a unit vector tangent to the intersection. For very small s, the term $s\mathbf{r}'(0)$ in (1) determines the orientation of the intersection and we choose the sign of $\mathbf{t}$ so as to maintain the orientation when s is positive. Specifically, let $\mathbf{r}'_{n-1}$ denote the derivative at the $(n-1)$-th point on the intersection. After $\mathbf{t}$ is computed for the n-th point, if $\mathbf{r}'_{n-1} \cdot \mathbf{t} < 0$, then we replace $\mathbf{t}$ with $-\mathbf{t}$.

For $m = 2$, the unique solution of (6) and the choice $\alpha_2 = 0$ leads to a unique vector $\mathbf{r}''(0)$. Then with κ the positive square root of $\mathbf{r}''(0) \cdot \mathbf{r}''(0)$, we have $\mathbf{r}''(0) = \kappa \mathbf{n}$, where $\mathbf{n}$ is the unit principle normal to the intersection.

Finally, by taking $\alpha_3 = -\kappa^2$, we have obtained the first three derivatives of $\mathbf{r}$ related as in (3).

3.3 Parametric definition

Next suppose that points on the surfaces are given in terms of parameters (u_k, v_k), $k = 1, 2$:

$$(x_k, y_k, z_k) = (G_{1,k}(u_k, v_k), G_{2,k}(u_k, v_k), G_{3,k}(u_k, v_k))$$

where the $G_{j,k}$ are given smooth functions. The intersection is defined by $G_{j,1}(u_1, v_1) = G_{j,2}(u_2, v_2)$, $j = 1, 2, 3$, a system of three equations in four unknowns. Once the unknowns have been determined as functions of s, points $\mathbf{r}(s)$ on the intersection are obtained by direct evaluation:

$$\mathbf{r}(s) = (G_{1,k}(u_k(s), v_k(s)), G_{2,k}(u_k(s), v_k(s)), G_{3,k}(u_k(s), v_k(s))) \quad (8)$$

where k is either 1 or 2.

Let $\mathbf{R}$ be the vector with four components defined by $\mathbf{R} = (u_1, v_1, u_2, v_2)$, and set

$$F_j(\mathbf{R}) = G_{j,1}(u_1, v_1) - G_{j,2}(u_2, v_2), \quad j = 1, 2, 3. \quad (9)$$

Then $F_j(\mathbf{R}(s)) \equiv 0$, and the Taylor expansion of $F_j(\mathbf{R}(s))$ is

$$F_j(\mathbf{R}(s)) = F_j(\mathbf{R}(0)) + s\nabla F_j \cdot \mathbf{R}'(0) \quad (10)$$
$$+\frac{s^2}{2}[\nabla F_j \cdot \mathbf{R}''(0) + \mathbf{R}'(0) \cdot H_{F_j} \cdot \mathbf{R}'(0)] + \cdots, \quad j = 1, 2, 3,$$

so that if $\mathbf{Q}$ is a solution of (9), then

$$\nabla F_j(\mathbf{Q}) \cdot \mathbf{R}^{(m)}(0) = B_{j,m}, \quad j = 1, 2, 3. \quad (11)$$

If the set of three gradients is linearly independent, then the general solution of (11) is

$$\mathbf{R}^{(m)} = \alpha_m \mathbf{T} + \beta_m \nabla F_1 + \gamma_m \nabla F_2 + \delta_m \nabla F_3, \quad (12)$$

where $\mathbf{T}$ is a unit vector orthogonal to the three gradients, and α_m is arbitrary.

Comparing equations (4), (5), and (7) with (10), (11), and (12), respectively, one sees that the only difference between the implicit and the parametric determination of the intersection is (a) the number of components of the vectors and (b) the number of equations. Thus, a general numerical method for one also applies to the other with minor modifications. In our Fortran implementation of the implicit formulation, this is accomplished by changing the size of some arrays and including the evaluation of points on the intersection with (8). Moreover, it should be evident that the method generalizes to tracing the intersection of $n-1$ smooth hypersurfaces in n-dimensional space assuming transversal intersections.

3.4 Newton Approximation

Given an initial point $\mathbf{p}_0$ near the curve, we find a point $\mathbf{q}$ on the curve by generating a sequence of points $\mathbf{p}_1, \mathbf{p}_2, \cdots \to \mathbf{q}$. We set $f_j(\mathbf{r}(s))) = 0$, $\mathbf{r}(0) = \mathbf{p}_k$ and $s\mathbf{r}'(0) = \Delta_k$ in (4), and neglect the terms with higher powers of s to get Newton's method. Thus we solve

$$\nabla f_j(\mathbf{p}_k) \cdot \Delta_k = -f_j(\mathbf{p}_k), \quad j = 1, 2. \tag{13}$$

Equation (13) is the same as equation (5). When the pair of gradients is linearly independent, the general solution is

$$\Delta_k = \alpha_k \mathbf{t} + \beta_k \nabla f_1(\mathbf{p}_k) + \gamma_k \nabla f_2(\mathbf{p}_k), \tag{14}$$

and the values of β_k and γ_k are determined uniquely. Because $\mathbf{t}$ is orthogonal to both surfaces, a change of $\mathbf{p}_k$ in the direction of $\mathbf{t}$ changes the values of f_j only negligibly, and we set $\alpha_k = 0$, thereby obtaining a unique solution for Δ_k. We then set $\mathbf{p}_{k+1} = \mathbf{p}_k + \Delta_k$.

Once the point $\mathbf{q}$ is found with acceptable accuracy, the approximation of $\mathbf{r}(s)$ with $\mathbf{r}(0) = \mathbf{q}$ is determined as described above.

3.5 Step Length

We use the higher order derivatives of $\mathbf{r}$ to estimate the accuracy of the low-order terms in the Taylor approximation. With this estimate, a step size is chosen such that the contribution of the second and third order terms together is at most 1/5 of the first order term. That is, we require that both

$$\|s^2\mathbf{r}''(0)/2\| \qquad \text{and} \qquad \|s^3\mathbf{r}'''(0)/6\|$$

are smaller than $\|s\mathbf{r}'(0)\|/10 = |s|/10$. For an example see Section 3.7 below. Since the step sizes could become arbitrarily small, a minimum step size is specified also.

3.6 Transformations of the Equations

The intersection of f_1 and f_2 is also the intersection of

$$\tilde{f}_1 = a_{1,1}f_1 + a_{1,2}f_2 \qquad \text{and} \qquad \tilde{f}_2 = a_{2,1}f_1 + a_{2,2}f_2$$

where $a_{j,k}$ are constants satisfying $a_{1,1}a_{2,2} - a_{1,2}a_{2,1} \neq 0$. Thus we can solve the equivalent system

$$\nabla \tilde{f}_j(\mathbf{q}) \cdot \mathbf{r}^{(m)}(0) = a_{j,1}b_{1,m} + a_{j,2}b_{2,m} \quad j = 1, 2,$$

where the $b_{j,m}$ are as before.

By choosing the constants $a_{j,k}$ suitably, we can, for example, formulate equivalent systems in which ∇f_1 and ∇f_2 are orthonormal or some of the intrinsic curve parameters, such as curvature radius, appear explicitly on the right side. This shifts the programming work to finding proper constants. Moreover, some of these choices parallel an algebraic approach to finding a local approximation at a singular curve point, as explained in Section 4.

3.7 Implementation

We have implemented the numerical tracing procedure in Fortran. Figures 3.1 through 3.8 show some examples of curve traces that were produced with this program and a standard graph utility under Berkeley Unix. The plane curves have been traced as the intersection of $f(x, y) = 0$ with $z = 0$, without any program modifications. As described further in the appendix, the linear system is solved using singular value decomposition [11, 23]. This approach is numerically very stable and increases the reliability in near-singular cases considerably.

At certain singularities, e.g., for the nodal singularity in Figure 3.5, the curve orientation $\nabla f_1 \times \nabla f_2$ reverses. This is a global property that depends on how the curve branches intersect at the singularity. If one were to determine its presence in this way, a complete analysis of the singularity would be required. To avoid this, we have added a heuristic that reverses the tracing direction whenever the oriented tangent changes by more than a maximum angle, say 90 degrees. In consequence, a cuspidal singularity cannot be traced with this algorithm.

In our experience, nodal singularities cause no problems as long as the tangent directions of the intersecting branches are sufficiently separated. Many tacnodes are also handled reliably, e.g., Figure 3.4. However, there are situations where branches may be confused. For example, both the curve $C_1 : y^2 - x^4 - y^4 = 0$ (Figure 3.7) and the curve $C_2 : y^2 - x^6 - y^6 = 0$ (Figure 3.8) are traced as if they had two real components meeting tangentially at the origin. While this is correct for C_1, it is not correct for C_2, since C_2

consists of a single real component with two branches at the origin, each having an inflection at the singularity. Note that the tangent computation of Section 4.1 or the singularity analysis of [19] does not suffice to distinguish the two cases.

The table below shows a short sample trace of $z + y^2 - x^3 \cap z + x^2$. The curve is shown graphically in Figure 3.2. The initial point estimate is $(0.2, 0.2, -0.1)$. The step length is determined adaptively as described. In addition to point coordinates, both the next step length and the number of Newton iterations needed to determine the point to within 10^{-10} are shown. For simplicity, only 5 decimals are given. At this singularity the orientation reverses and is reflected in the change of sign of the step length.

Point			Iter	Next Step
(+0.199682	+0.218711	−0.039873)	3	−0.27637
(+0.015686	+0.015809	−0.000246)	2	−0.19340
(−0.124761	−0.116720	−0.015565)	2	+0.16081
(−0.245628	−0.213339	−0.060333)	2	+0.15031
(−0.358096	−0.286903	−0.128233)	2	+0.15308

Since the third derivative $\mathbf{r}'''$ is not necessarily perpendicular to $\mathbf{r}'$, the point distance does not always correspond to the step length.

4 Singular Curve Points

Consider now the intersection curve when the surfaces are given implicitly by f_1 and f_2. At a singularity p, the Taylor expansion of $\mathbf{r}$ does not exist in the ordinary sense. Nevertheless, System (5) remains formally valid and can be used to determine approximants to $\mathbf{r}$ at p. This fact is less attractive than one might suspect at first, since the equations no longer are linear and, thus, become more difficult to solve. A point p is *singular* on $\mathbf{r}(s)$ for one of the following reasons:

1. The gradients ∇f_1 and ∇f_2 are linearly dependent.

2. One of the gradients, say ∇f_2 is zero, but the other is not.

3. Both gradients ∇f_1 and ∇f_2 are zero.

4.1 Tangents at Singular Points

We consider Case 1, i.e., linearly dependent surface gradients. From Section 3.6 it follows that this case is in substance the same as Case 2, and we demonstrate how the familiar tangent cone construction corresponds to an elementary simplification of the Equation System (5).

When the gradients are linearly dependent, the tangent planes of f_1 and f_2 are the same. We assume without loss of generality that the point p is the origin and $\nabla f_1 = (0,0,1)$. Therefore, we may write

$$f_1 = z + \bar{f}_1 = 0$$
$$f_2 = \mu z + \bar{f}_2 = 0$$

where the polynomials $\bar{f}_1$ and $\bar{f}_2$ consist of terms of degree 2 or higher.

Now the intersection of f_1 and f_2 is also the intersection of f_1 and $f_3 = f_2 - \mu f_1$. We determine the curve tangent(s) from f_1 and f_3. The terms of lowest order in f_3 comprise a homogeneous form F that approximates the surface $f_3 = 0$ in the neighborhood of the origin and has degree 2 or higher. $F(x, y, z) = 0$ is a cone with the origin at its vertex. It intersects the plane tangent to $f_1 = 0$ only at the origin or in a set of lines through the origin that are tangent to the branches of $\mathbf{r}$, the intersection of f_1 and f_2.

It is possible that F is divisible by z. In that case the computation must be iterated; i.e., we must determine a f_4 by subtracting from f_3 a multiple of $z^k f_1$, where k is suitably chosen. [16] proves that this computation terminates.

We determine the tangents to the intersection at the origin by substituting $z = 0$ in F. This yields the homogeneous polynomial $F(x, y, 0)$ in two variables. The roots of $F(x, y, 0)$ are $(0, 0)$ and $(\lambda u, \lambda v)$ where not both u and v are zero and $\lambda \neq 0$. The root (0,0) is an improper solution for G and is excluded. If there are no other real roots, then the cone intersects the plane $z = 0$ only in the origin, a case that does not arise when tracing a curve branch.

For every other real root $(\lambda u, \lambda v)$ we obtain a corresponding tangent vector $\mathbf{r}' = (\lambda u, \lambda v, 0)$ to $\mathbf{r}$ at the origin. Here λ is chosen such that the vector has length 1.

We demonstrate by example that this tangent computation is equivalently done by elementary manipulation of the equations of System (5). The deeper reason for this is further clarified below and rests on the correspondence of the Taylor series at regular curve points with formal power series expansions of r at singularities.

Example

Consider the intersection of the two cylinders $f_1 = x^2 + z^2 + 2z = 0$ and $f_2 = y^2 + z^2 + 4z = 0$ which is irreducible and has a nodal singularity at the origin, as shown in Figure 3.1. The curve is equivalently the intersection of f_1 with the elliptic cone $f_3 = f_2 - 2f_1 = y^2 - 2x^2 - z^2$. For this cone $F = f_3$. Therefore the tangents at the origin are given by the roots of $y^2 - 2x^2$, i.e., they are the lines $(\lambda, \sqrt{2}\lambda, 0)$ and $(-\lambda, \sqrt{2}\lambda, 0)$.

Next, when we formulate the equations of System (5) for f_1 and f_2 and write $(x(s), y(s), z(s))$ for $\mathbf{r}(s)$, we obtain at the origin

$$\begin{aligned} 2z'(s) &= 0 \\ 4z'(s) &= 0 \\ 2z'' &= -2x'^2 - 2z'^2 \\ 4z'' &= -2y'^2 - 2z'^2 \end{aligned}$$

By subtracting the third equation twice from the fourth and dividing by two, we obtain the equation

$$0 = 2x'^2 - y'^2 + z'^2$$

Note the similarity between this equation and the tangent cone of f_3. Thus, solving System (5) for $\mathbf{r}'$ is equivalent to determining the tangent directions from f_1 and f_3.

4.2 Algebraic Correspondence

We assume that $\mathbf{r}(0)$ is the origin and write

$$\mathbf{r}(s) = \sum_{i \geq 1} (a_i, b_i, c_i) s^i$$

where (a_i, b_i, c_i) is a vector, e.g., [25, Ch. IV.2, V.5]. The formal derivative of $\mathbf{r}$ by s is defined as

$$\mathbf{r}'(s) = \sum_{i \geq 0} (a_{i+1}, b_{i+1}, c_{i+1})(i+1) s^i$$

The power series must satisfy identically $f(\mathbf{r}(s)) = 0$. Substituting the series of $\mathbf{r}(s)$ into f and collecting terms, we obtain a formal series

$$\sum_{i\geq 1} K_m s^m \equiv 0$$

This leads to a system of equations

$$K_m = 0, \qquad m = 1, 2, 3, \ldots$$

where K_m is the coefficient of s^m in the resulting series. A similar system of equations is obtained for $g(\mathbf{r}(s)) = 0$. Because the formal derivative above has all the familiar properties of derivatives, these equations are formally the same as System (5).

Because of this algebraic correspondence, it is possible to approximate the curve at a singular point by formulating the system of equations as before and solving it for the unknown coefficients. In contrast to the nonsingular case, however, the system no longer is linear and thus is more difficult to solve. We explain the procedure by an example:

We consider the intersection of the surfaces $f_1 = z + y^2 - x^3$ and $f_2 = z + x^2$ with a nodal singularity at the origin, as shown in Figure 3.2. We set

$$\begin{aligned}
x(s) &= a_1 s + a_2 s^2 + a_3 s^3 + \cdots \\
y(s) &= b_1 s + b_2 s^2 + b_3 s^3 + \cdots \\
z(s) &= c_1 s + c_2 s^2 + c_3 s^3 + \cdots
\end{aligned}$$

where $(a_1, b_1, c_1) = \mathbf{r}'(0)$, $(a_2, b_2, c_2) = \mathbf{r}''(0)/2$, and so on. The equations of System (13), or equivalently, of System (5), are thus

$$\begin{aligned}
c_1 &= 0 \\
c_2 + b_1^2 &= 0 \\
c_2 + a_1^2 &= 0 \\
c_3 + 2a_1 a_2 &= 0 \\
c_4 + 2b_1 b_3 + b_2^2 - 3a_1^2 a_2 &= 0 \\
c_4 + 2a_1 a_3 + a_2^2 &= 0 \\
&\vdots
\end{aligned}$$

As before, the system is underconstrained. It is possible to choose the independent quantities such that one of the series is $\pm s^k$ [25]. Here s need not correspond to arc length. We choose $c_2 = -1$ and $c_3 = c_4 = \cdots = 0$. Then the following two solutions are obtained:

$$\begin{aligned} x(s) &= s \\ y(s) &= s + \frac{s^2}{2} - \frac{s^3}{8} \pm \cdots \\ z(s) &= -s^2 \end{aligned}$$

and

$$\begin{aligned} x(s) &= s \\ y(s) &= -s - \frac{s^2}{2} + \frac{s^3}{8} \mp \cdots \\ z(s) &= -s^2 \end{aligned}$$

The series correspond to the local parameterizations of the two intersecting branches. For remarks about their convergence see, e.g., [24, p. 52].

Because the equations are nonlinear, this approach is difficult to implement. The degree of the equations depends on the order of the singularity. In the simplest cases this is two. However, higher order singularities can occur that may make it difficult to solve the equations and to identify subsequently a solution that parameterizes the traversed path.

5 Plane Curves

We now consider tracing a segment of the plane algebraic curve $f(x, y) = 0$, beginning at an initial point (x_0, y_0) at which tracing commences in a specified direction. For simplicity, we assume that the initial point is not singular. With this assumption, the trace direction is simply specified as *positive*, following the tangent vector $(-f_y, f_x)$, or *negative*, tracing in the opposite direction. If the initial point is singular, a more complicated specification procedure is required that identifies the intended branch and a direction on it. Such specifications can be worked out without difficulty, based on the desingularization techniques described below. See also [13] for a discussion of this problem in the context of solid modeling.

5.1 . Desingularization

Desingularization of plane curves is based on the following classical theorem, proved by Riemann and Cayley:

Theorem: Every plane curve can be birationally transformed into a curve devoid of singularities.

Among the different proofs of the theorem are constructive versions that derive the needed birational transformation from a sequence of simple quadratic transformations, e.g., [1, 24, 25]. Two transformations are needed:

$$\begin{aligned} T_1: \quad x' &= x \\ y' &= y/x \\ T_2: \quad x' &= x/y \\ y' &= y \end{aligned}$$

The inverse transformations are, respectively, $x = x'$, $y = x'y'$, and $x = x'y'$, $y = y'$. The basic properties of transformation T_1 can be summarized as follows:

1. All points (x, y) with $x \neq 0$ are mapped 1-1 to the x'-y' plane.
2. All points $(0, y)$ with $y \neq 0$ are mapped to infinity.
3. As we approach the origin on a branch, the limit of the image points is the image of the origin on the branch. This limit depends on the direction of approach, hence the pencil of directions through the origin, except the y-axis, are mapped to finite points on the y'-axis.

In particular, T_1 maps irreducible curves to irreducible curves. The line $x' = 0$ is called the *exceptional* line of T_1. The properties of T_2 are analogous. The exceptional line of T_2 is $y' = 0$.

In intuitive terms, the transformations separate curve branches that intersect with different tangent directions. This is plausible since the line $y - mx = 0$ through the origin is mapped to the line $y' - m = 0$ that intercepts the y' axis at distance m from the origin. Moreover, branches that are in higher order contact, such as tacnodes, are mapped to singularities in the x'-y' plane at which the contact order is reduced. Finally, the order of a nonlinear branch through the origin is also reduced. The latter two facts are not easily seen, as they depend on structural properties not readily apparent from the graph of f and the elementary concepts such as tangent direction, curvature, etc. Nevertheless, given a suitable measure for the complexity of a singular point, it can be shown that every application of T_1 or T_2 simplifies the complexity of the point, so that the topology of the singularity is eventually resolved into a tree structure, each of whose leaves corresponds to a nonsingular curve branch. For example, [1] defines such a measure based on the structure of the order form, [25] uses a measure related to the curve genus, whereas [24] uses the intersection multiplicity of the branch with the polar form as a measure of complexity.

Since T_1 maps a branch with the y-axis as tangent to infinity, such a branch must be desingularized using T_2. Likewise, T_1 must be applied to branches each of whose tangent is the x-axis. For the intermediate tangent positions we choose T_1 if the slope of the tangent has magnitude 1 or less; otherwise T_2 is chosen. Figures 5.1 through 5.3 show some examples of curve desingularization. It will be noted that for more complex singularities the transformations have to be applied repeatedly. Moreover, since we trace a particular branch, it will not concern us if a different branch is mapped to infinity.

5.2 Tracing with Desingularization

The tracing method consists of a numerical part used to trace noncritical parts of the curve. Upon detecting an impending singularity, the branch is transformed by T_1 or T_2, and the transformed branch is traced. This secondary trace continues until we are safely past the singularity, at which point tracing returns to the original curve. The procedure is recursive when the transformed branch in turn is singular. Its central steps are as follows:

1. When the system determinant approaches zero, locate the singular point q expected to lie close to p by the iteration described below. Record the order of the singularity, as obtained by this iteration procedure.

2. Translate the coordinate system to bring the singularity to the origin. Eliminate low order terms as required by the order of the singularity.

3. Depending on the tangent direction at the singularity, apply T_1 or T_2 to obtain the transformed curve g. Establish on g the point p' corresponding to p and the appropriate direction of traversal.

4. Trace g until the exceptional line is reached. Then trace an equal number of steps beyond that point.
5. Map the points traced on g back to f and return to tracing f after establishing the correct traversal direction.

Our implementation applies designularization recursively as needed.

5.3 Locating the Singularity

When the system determinant approaches zero at P_0, we locate a singularity in the vicinity. The singular point is defined as the common intersection of $f = 0$, $f_x = 0$ and $f_y = 0$. Using a Newton iteration, we construct a sequence of points $P_0, P_1, P_2, \ldots$ converging to the singularity. Let $P_{i+1} = P_i + (\delta_x, \delta_y)$. Then we solve the linear system

$$\begin{pmatrix} f_x & f_y \\ f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} -f \\ -f_x \\ -f_y \end{pmatrix}$$

This is an overconstrained system and may be solved using a least-squares approach, thus solving

$$A^T A \Delta = A^T b,$$

where A is the coefficient matrix of the overconstrained system, Δ is the vector (δ_x, δ_y), and b is the right hand side.

If the singularity has order higher than 2, then the two by two matrix $A^T A$ does not have full rank, and the three partials f_{xx}, f_{xy} and f_{yy} also vanish. In this case we must augment A by higher order derivatives. In particular, for a third order singularity we have

$$A = \begin{pmatrix} f_x & f_y \\ f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \\ f_{xxx} & f_{xxy} \\ f_{xxy} & f_{xyy} \\ f_{xyy} & f_{yyy} \end{pmatrix} \qquad b = -\begin{pmatrix} f \\ f_x \\ f_y \\ f_{xx} \\ f_{xy} \\ f_{yy} \end{pmatrix}$$

Since possibly only some of the next order partials vanish, we proceed adaptively as follows:

Whenever the partial h in the matrix A vanishes, then A is augmented by the row (h_x, h_y) and b the entry $-h$.

In this manner a matrix $A^T A$ of full rank is obtained.

If the partial $f_{x^j y^k}$ vanishes at the singularity p, it follows that f cannot contain the term $cx^j y^k$, $c \neq 0$, after the origin has been translated to p. Since the translation of f to the origin incurs numerical errors, it is possible that f contains such a term with a very small coefficient c. Such terms must be eliminated.

5.4 Direction of Traversal

At nonsingular points, we give a standard orientation by the tangent vector $(-f_y, f_x)$, as shown in Figure 5.4. The orientation is not intrinsic since $-f(x, y) = 0$ results in the opposite orientation of the curve. At a singularity, curve segments locally belonging to the same analytic branch may be oriented in an opposite direction, as shown in Figure 5.5. So, we establish a relationship between the orientation of the curve f and the orientation of its proper transform g.

Whether the branch orientation reverses at a singularity depends on the structure of the singular point. Since the gradient $\nabla f = (f_x, f_y)$ always points away from the area of negative points (a, b), i.e., points such that $f(a, b) < 0$, the branch orientation reverses precisely when this branch intersects an even number of other branches. Two examples, Figures 5.6 and 5.7, show the curve in the neighborhood of the singularity as well as a schematic diagram of the topological structure of the singularity.

We now quantify the correspondence between the orientation of f and its proper transform g and derive a simple method for detecting orientation reversal without having to analyze the topological structure of the singularity in detail. Let $p = (a_0, b_0)$ be a nonsingular point of f, where $a_0 \neq 0$. Let

$$x(s) = a_0 + a_1 s + a_2 s^2 + \ldots$$

$$y(s) = b_0 + b_1 s + b_2 s^2 + \ldots$$

be the place of f centered at p. The place defines a branch orientation by increasing s that need not agree with the standard orientation $(-f_y, f_x)$. Centered at the corresponding point $p_1 = (a_0, b_0/a_0)$, the transformed curve

g has the place

$$x_1(s) = x(s)$$
$$y_1(s) = c_0 + c_1 s + c_2 s^2 + \ldots$$

Since $x(s) = x_1(s)$, the curve and its transform are oriented the same way. Moreover, since $y_1(s) = y(s)/x(s)$, we divide the two power series to obtain

$$c_0 = b_0/a_0$$

$$c_1 = (b_1 a_0 - a_1 b_0)/a_0^2$$

and so on. Now p and p_1 are not singular. Consequently, the Taylor series exists, a_1 is proportional to $-f_y$ and $-g_y$, b_1 is proportional to f_x, and c_1 is proportional to g_x. Thus, the sign of the proportionality factor α relates the orientation of the Taylor series with the standard orientation. Therefore, given the direction of tracing f, we obtain the corresponding tracing direction of g from

$$g_y = \alpha f_y$$
$$g_x = \alpha(x f_x + y f_y)/x^2$$

Conversely, given the tracing direction of g, we obtain the corresponding tracing direction of f in the same way.

In consequence, the following procedure is used to maintain a consistent tracing direction through singularities:

1. We traverse f in the direction $u(-f_y, f_x)$, where $u = 1$ or $u = -1$.

2. When approaching a singular point, the proper transform g of f is calculated. Let p be a point on f traversed before the singularity, and let p_1 be the corresponding point on g. The partials of f and g are evaluated at these points, and the factor α of proportionality determined as described above.

3. If $\alpha > 0$, the transform g is traversed in the direction $u(-g_y, g_x)$; otherwise, it is traversed in the opposite direction.

The same traversal correlation is established when leaving the vicinity of the singularity, reestablishing the proper traversal direction on f from the traversal direction on g.

5.5 Implementation

We have implemented the algorithm on a Symbolics 3650 Lisp machine and traced the curves shown in Figures 5.8 through 5.15. In our experience with the program, it is possible to trace through complex singularities. A problem for the present implementation is locating the singularity accurately. For example, locating the cuspidal singularity of the family of curves $y^2 - x^{2m+1} = 0$ becomes increasingly more difficult as m grows. Another problem arises when a curve is almost singular, as in the case of the family of curves $y^2 - x^2 - x^3 - \epsilon = 0$. For very small values of ϵ the curve has very high curvature in the vicinity of the origin and appears to be singular.

Acknowledgements

We thank Prof. Abhyankar for various insightful discussions in algebraic geometry.

6 References

1. Abhyankar, S., (1983), Desingularization of Plane Curves, *Proc. of Symposia in Pure Mathematics*, AMS, 40, 1, 1-45.

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

3. Abhyankar, S., and Bajaj, C., (1987b), Automatic Rational Parameterization of Curves and Surfaces II: Cubics and Cubicoids, *Computer Aided Design*, 19, 9, 499-502.

4. Abhyankar, S., and Bajaj, C., (1987c), Automatic Rational Parameterization of Curves and Surfaces III: Algebraic Plane Curves, Tech. Rept. 619, Comp. Science, Purdue University.

5. Abhyankar, S., and Bajaj, C., (1987d), Automatic Rational Parameterization of Curves and Surfaces IV: Algebraic Space Curves, Tech. Rept. 703, Comp. Science, Purdue University.

6. Boehm, W., Farin, G., and Kahmann, J., (1984), A Survey of Curve and Surface Methods in CAGD, *Computer Aided Geometric Design*, 1, 1-60.

7. Cohen, E., Lyche, T., and Riesenfeld, R., (1980), Discrete B-splines and Subdivision Techniques in Computer Aided Geometric Design and Computer Graphics, *Computer Graphics and Image Processing*, 14, 87 - 111.

8. Dongarra, J., Moler, C., Bunch, J., Stewart, G., (1979), *Linpack User's Guide*, SIAM Publications, Philadelphia

9. Farouki, R., (1986), Trimmed Surface Algorithms for the Evaluation and Interrogation of Solid Boundary Representations, *IBM J. Res. and Dev.*, 31.

10. Franklin, P., (1944), *Methods of Advanced Calculus*, McGraw-Hill

11. Golub, G., and Van Loan, C., (1985), *Matrix Computations*, Johns Hopkins University Press, Baltimore

12. Hoffmann, C., (1987), Algebraic Curves, Tech. Rept. 675, Comp. Science, Purdue University.

13. Hoffmann, C., and Hopcroft, J., (1987), Geometric Ambiguities in Boundary Representations, *Computer Aided Design*, 19, 3, 141 - 147.

14. Levin, J., (1979), Mathematical Models for Determining the Intersections of Quadric Surfaces, *Computer Graphics and Image Processing*, 11, 73 - 87.

15. Montaudouin, Y., Tiller, W., and Vold, H., (1986), Applications of Power Series in Computational Geometry, *Computer Aided Design*, 18, 10, 514 - 524.

16. Mora, F., (1982) An Algorithm to Compute the Equations of Tangent Cones, *EOROCAM '82*, Springer Lect. Notes in Comp. Science 144, 158–165

17. Pratt, M., (1986), Parametric Curves and Surfaces as used in Computer Aided Design, *The Mathematics of Surfaces*, ed. J. Gregory, Oxford University Press, 19–46.

18. Ocken, S., Schwartz, J., and Sharir, M., (1983), Precise Implementation of CAD Primitives using Rational Parameterizations of Standard Surfaces, *Planning, Geometry and Complexity of Robot Motion*,

19. Owen, J., Rockwood, A., (1987) Intersection of General Implicit Surfaces, in *Geometric Modeling: Algorithms and Trends*, G. Farin, ed., SIAM Publications, Philadelphia

20. Pratt, M., Geisow, A., (1986), Surface/Surface Intersection Problems, *The Mathematics of Surfaces*, ed. J. Gregory, Oxford University Press, 117 - 142.

21. Requicha, A., and Voelcker, H., (1983), Solid Modeling: Current Status and Research Directions, *IEEE Comp. Graphics and Applic.*, 3, 25 - 37.

22. Snyder, V., and Sisam, C., (1914), *Analytic Geometry of Space*, Henry Holt and Company, New York.

23. Stewart, G., (1973), *Introduction to Matrix Computations*, Academic Press

24. van der Waerden, B., (1938), *Einführung in die algebraische Geometrie*, Springer Verlag, 2nd edition, 1973

25. Walker, R., (1950), *Algebraic Curves*, Springer Verlag 1978, New York

26. Wesley, M., Lozano-Perez, T., Liberman, L., Lavin, M., Grossman, D., (1980), A Geometric Modeling System for Automated Mechanical Assembly, *IBM J. Res. and Dev.*, 24, 64 - 74.

A Appendix: Computational Details

We describe in more detail the derivation of the quantities $b_{i,m}$, $\quad i = 1, 2$, $B_{j,m}$, $\quad j = 1, 2, 3$ and the use of the singular value decomposition to solve the linear system of Section 3.

A.1 Derivation of the $b_{i,m}$ and $B_{j,m}$

The expressions for $b_{i,m}$ are developed from the Taylor expansion of f_1 and f_2 of Section 3.2. For f_1 we obtain

$$f_1(x, y, z) = f_1(x_0 + \Delta x, y_0 + \Delta y, z_0 + \Delta z) = \sum_{i,j,k} f_{i,j,k} \Delta x^i \Delta y^j \Delta z^k,$$

where

$$f_{i,j,k} = \frac{1}{i!j!k!}\frac{\partial^{i+j+k}}{\partial x^i \partial y^j \partial z^k} f_1(x_0, y_0, z_0).$$

We set $\Delta x = x's + x''s^2/2 + x'''s^3/6 + \cdots$, $\Delta y = y's + y''s^2/2 + \cdots$, etc. Then

$$(\Delta x)^2 = (x')^2 s^2 + x'x''s^3 + \cdots, \qquad (\Delta x)^3 = (x')^3 s^3 + \cdots,$$
$$\Delta x \Delta y = x'y's^2 + (x''y' + x'y'')s^3/2 + \cdots, \qquad \Delta x \Delta y \Delta z = x'y'z's^3 + \cdots,$$

and so on. Substituting into the Taylor's series for f_1 and equating to zero the coefficients of s^m, $m = 1, 2, 3$, we get the equations

$$f_{1,0,0}x' + f_{0,1,0}y' + f_{0,0,1}z' = 0,$$

$$\begin{aligned} f_{1,0,0}x'' + f_{0,1,0}y'' + f_{0,0,1}z'' \\ = -2[f_{2,0,0}(x')^2 + f_{0,2,0}(y')^2 + f_{0,0,2}(z')^2 \\ + f_{1,1,0}x'y' + f_{1,0,1}x'z' + f_{0,1,1}y'z'], \end{aligned}$$

$$\begin{aligned} f_{1,0,0}x''' + f_{0,1,0}y''' + f_{0,0,1}z''' \\ = -6[f_{2,0,0}x'x'' + f_{0,2,0}y'y'' + f_{0,0,2}z'z'' + f_{1,1,0}(x''y' + x'y'')/2 \\ + f_{1,0,1}(x''z' + x'z'')/2 + f_{0,1,1}(y''z' + y'z'')/2 \\ + f_{3,0,0}(x')^3 + f_{0,3,0}(y')^3 + f_{0,0,3}(z')^3 \\ + f_{2,1,0}(x')^2y' + f_{1,2,0}x'(y')^2 + f_{2,0,1}(x')^2z' \\ + f_{1,0,2}x'(z')^2 + f_{0,2,1}(y')^2z' + f_{0,1,2}y'(z')^2 + f_{1,1,1}x'y'z']. \end{aligned}$$

They are the equations

$$\nabla f_1 \cdot \mathbf{r}' = 0, \qquad \nabla f_1 \cdot \mathbf{r}'' = b_{1,2}, \qquad \nabla f_1 \cdot \mathbf{r}''' = b_{1,3}$$

of Section 3.1. The explicit form above is used for computing $b_{1,2}$ and $b_{1,3}$ in the program. A similar set of formulae is obtained for computing $b_{2,2}$ and $b_{2,3}$ when f_1 is replaced with f_2.

The expressions for $B_{j,m}$ are developed, in an analgous fashion, from the Taylor expansion of F_1, F_2 and F_3 of Section 3.3.

A.2 Singular Value Decomposition

Both Newton's method for refining a point estimate and the determination of the curve approximant entail solving a linear system

$$A^T \mathbf{w} = \mathbf{z},$$

For the implicit case A is a 3-by-2 matrix whose columns are the gradients of f_1 and f_2, and where $\mathbf{w}$ and $\mathbf{z}$ are column vectors of length 3 and 2, respectively. For the parametric case A is a 4-by-3 matrix whose columns are the gradients of F_1, F_2 and F_3, and where $\mathbf{w}$ and $\mathbf{z}$ are column vectors of length 4 and 3, respectively.

When the pair of gradients is linearly independent, then the general solution of this system was written in Section 3.2 as

$$\mathbf{w} = \alpha \nabla f + \beta \nabla g + \gamma \mathbf{t}$$

and in Section 3.6 as

$$\mathbf{w} = \alpha \nabla F + \beta \nabla G + \gamma \nabla H + \zeta \mathbf{t}.$$

This is not the general solution at a singularity where the pair of gradients is linearly dependent.

To treat all cases in a uniform way with a computationally stable process, we compute the singular value decomposition of A [11, 23]. (We linked the thoroughly tested routines of Linpack [8] to our program.) Thus, we factor A as $A = U\Sigma V^T$, where $U \in \mathbf{R}^{3\times 3}$ and $V \in \mathbf{R}^{2\times 2}$ for the implicit/implicit case are orthogonal matrices and $\Sigma \in \mathbf{R}^{3\times 2}$ is diagonal. For the parametric/parametric case $U \in \mathbf{R}^{4\times 4}$ and $V \in \mathbf{R}^{3\times 3}$ are orthogonal matrices and $\Sigma \in \mathbf{R}^{4\times 3}$ is diagonal. The system $A\mathbf{w} = \mathbf{z}$ now becomes

$$V\Sigma^T U^T \mathbf{w} = \mathbf{z},$$

and we write its solution as

$$\mathbf{w} = \alpha' \mathbf{U}_1 + \beta' \mathbf{U}_2 + \gamma' \mathbf{U}_3,$$

where $\mathbf{U}_j$ denotes the j-th column of U. Since the gradients ∇f_1 and ∇f_2 and the vector $\mathbf{t}$ are not generally orthonormal, and since the $\mathbf{U}_j$ are, the quantities α', β', and γ' differ from their counterpart in Section 3.1.

There are three cases:

(i) If the pair of gradients is linearly independent, then $\Sigma_{1,1} > 0$, $\Sigma_{2,2} > 0$, and the first two columns of U span the same space as the pair of gradients. In that case,

$$\alpha' = (\mathbf{V}_1{}^T \mathbf{z})/\Sigma_{1,1}, \qquad \beta' = (\mathbf{V}_2{}^T \mathbf{z})/\Sigma_{2,2},$$

and γ' is arbitrary.

(ii) If the pair of gradients is linearly dependent and at least one is nonzero, then $\Sigma_{1,1} > 0$, $\Sigma_{2,2} = 0$, and the first column of U spans the same space as the pair of gradients. If $\mathbf{V}_2{}^T\mathbf{z} \neq 0$, then there is no solution; otherwise

$$\alpha' = (\mathbf{V}_1{}^T\mathbf{z})/\Sigma_{1,1},$$

and β' and γ' are arbitrary.

(iii) If both gradients are zero, then so is Σ. If $\mathbf{z} \neq 0$, then there is no solution; otherwise α', β', and γ' are arbitrary.

This is now used as follows.

Newton's method. We always choose $\gamma' = 0$. In Case (ii), β' is also set to zero. In Case (iii), the initial guess is perturbed and the iteration restarted. Usually two or three iterations suffice. If the singular value decomposition is not recomputed at each iteration, the number of iterations typically doubles.

Finding the Approximant. The solutions to the linear systems are determined using the Frenet-Serret formulae [10, p. 107]:

$$\frac{d\mathbf{t}}{ds} = \kappa\mathbf{n}, \qquad \frac{d\mathbf{b}}{ds} = -T\mathbf{n}, \qquad \frac{d\mathbf{n}}{ds} = T\mathbf{b} - \kappa\mathbf{t},$$

where s is arc length, $\mathbf{t}$ is the unit tangent, $\mathbf{n}$ is the principle normal, $\mathbf{b}$ is the binormal, $\kappa = 1/\rho$ is curvature, and $T = 1/\tau$ is torsion. The vectors $\mathbf{t}$, $\mathbf{n}$, and $\mathbf{b}$ form an orthonormal triad with

$$\mathbf{n} = \mathbf{b} \times \mathbf{t}.$$

At a point $\mathbf{r}(s)$ on the curve, we have

$$\mathbf{r}'(s) = \mathbf{t}, \qquad \mathbf{r}''(s) = \frac{d\mathbf{t}}{ds} = \kappa\mathbf{n},$$

$$\mathbf{r}'''(s) = \frac{d}{ds}(\kappa\mathbf{n}) = \frac{d\kappa}{ds}\mathbf{n} + \kappa\frac{d\mathbf{n}}{ds} = \kappa'\mathbf{n} + \kappa T\mathbf{b} - \kappa^2\mathbf{t}.$$

In Case (i) we obtain $\mathbf{r}' = \gamma_1'\mathbf{U}_3$ using $\gamma_1' = \pm 1$. For the first point on the curve, the sign of γ_1' is an input parameter; for other points, the sign of γ_1' is chosen to be the sign of $\mathbf{r}'(0)^T\mathbf{U}_3$ at the previous point $\mathbf{r}'(0)$. To get $\mathbf{r}''(s)$, we use $\gamma_2' = 0$, so that $\mathbf{r}'$ and $\mathbf{r}''$ are orthogonal. The length of $\mathbf{r}''(s)$ gives the curvature κ. To get $\mathbf{r}'''(s)$, we choose $\gamma_3' = -\kappa^2$.

In Case (ii), we project $\mathbf{r}'(0)$ into the plane spanned by U_2, U_3, and then normalize the projection to get $\mathbf{r}'(s)$; an input vector is given if $k = 0$. For $\mathbf{r}''$, we choose β_2' and γ_2' to make $\mathbf{r}''$ and $\mathbf{r}'$ orthogonal; $\mathbf{r}'''$ is chosen as above.

In Case (iii), we return to the preceding point and double the computed step length.

Figure 3.1
Cylinder - Cylinder Intersection
$x^2 + z^2 + 2z = 0 \cap y^2 + z^2 + 4z = 0$

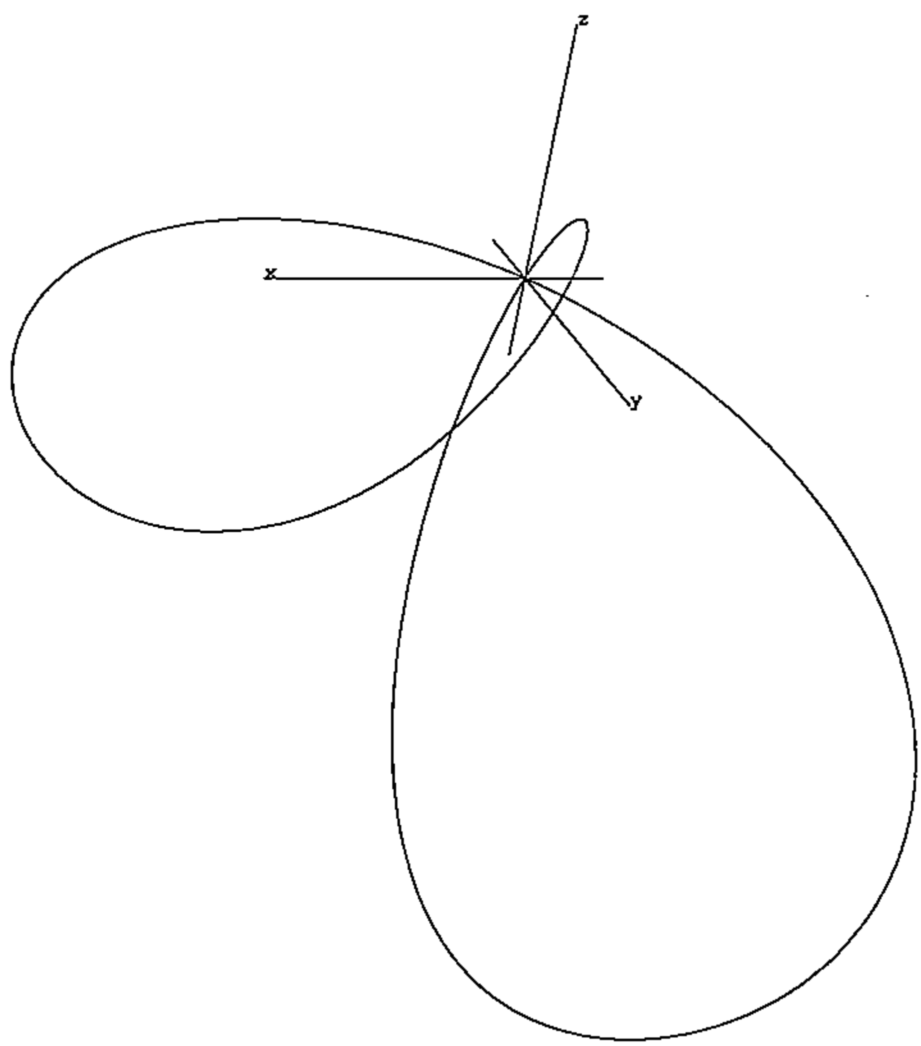


Figure 3.2
Nodal Singularity
$z + y^2 - x^3 = 0 \cap z + x^2 = 0$

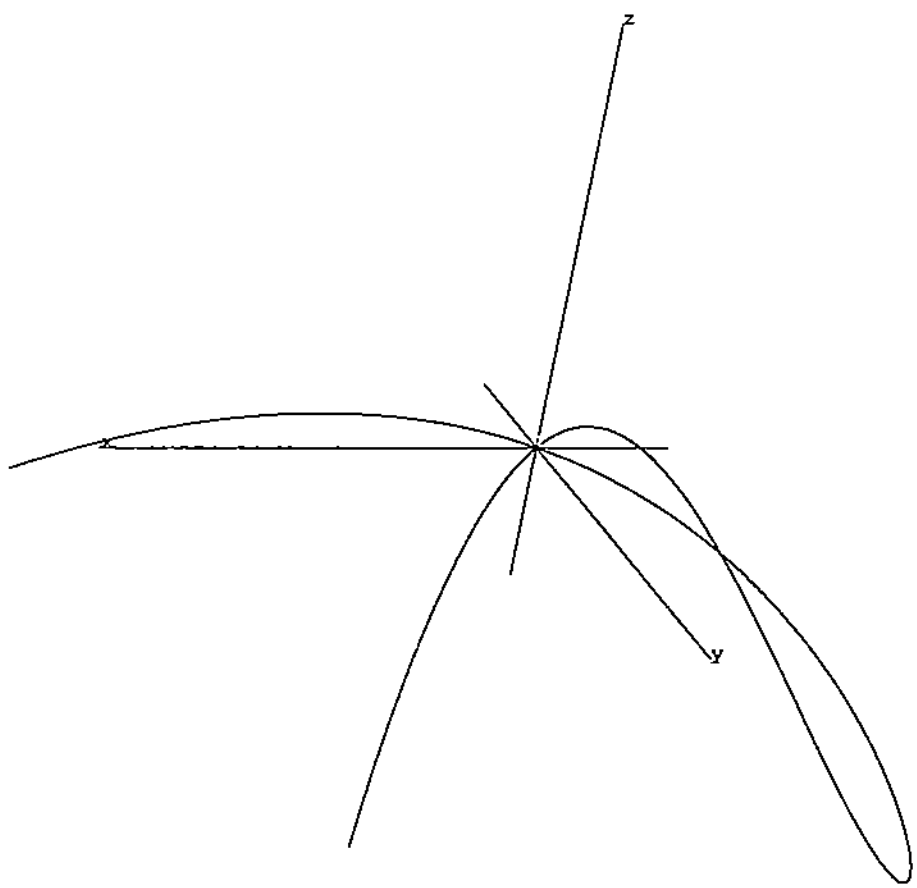


Figure 3.3
Tacnode Singularity
$z + x^4 + y^4 = 0 \cap z + y^2 = 0$

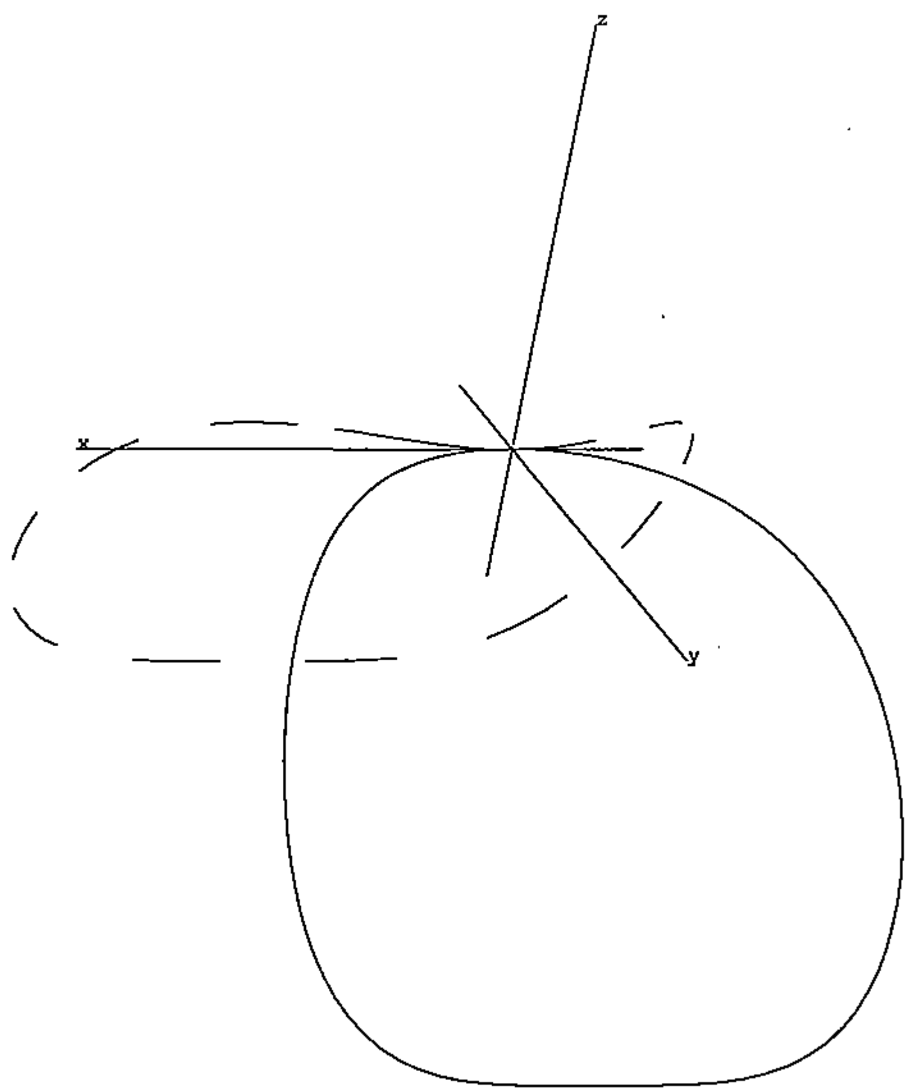


Figure 3.4
Tacnode and Nodal Singularities
$z - 2x^4 - y^4 = 0 \cap z - 3x^2y + y^2 - 2y^3 = 0$

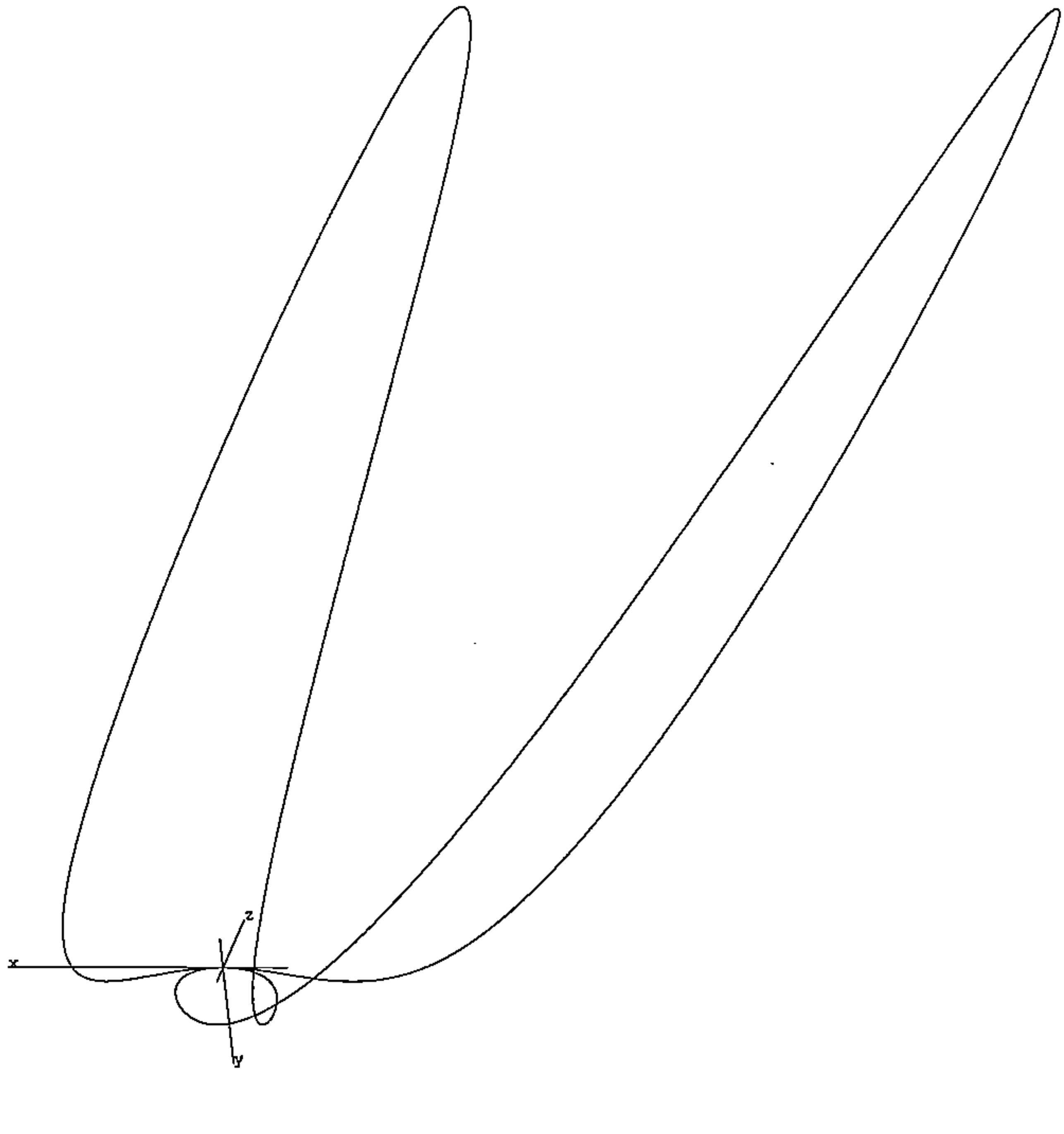


Figure 3.5
Projection of Figure 3.2 onto the Plane $z = 0$
$y^2 - x^2 - x^3 = 0$

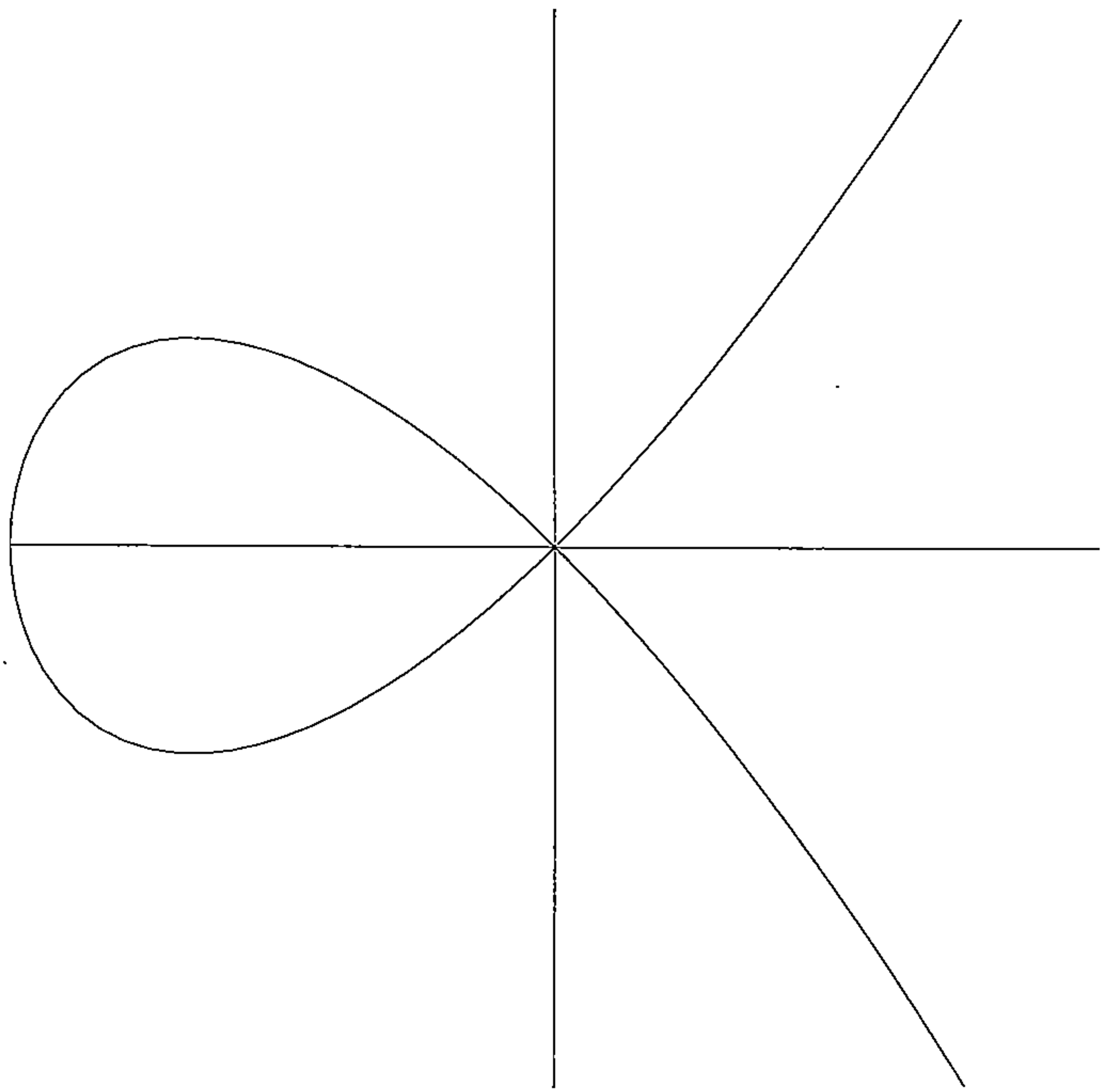


Figure 3.6
Projection of Figure 3.4 onto the Plane $z = 0$
$2x^4 - 3x^2y + y^2 - 2y^3 + y^4 = 0$

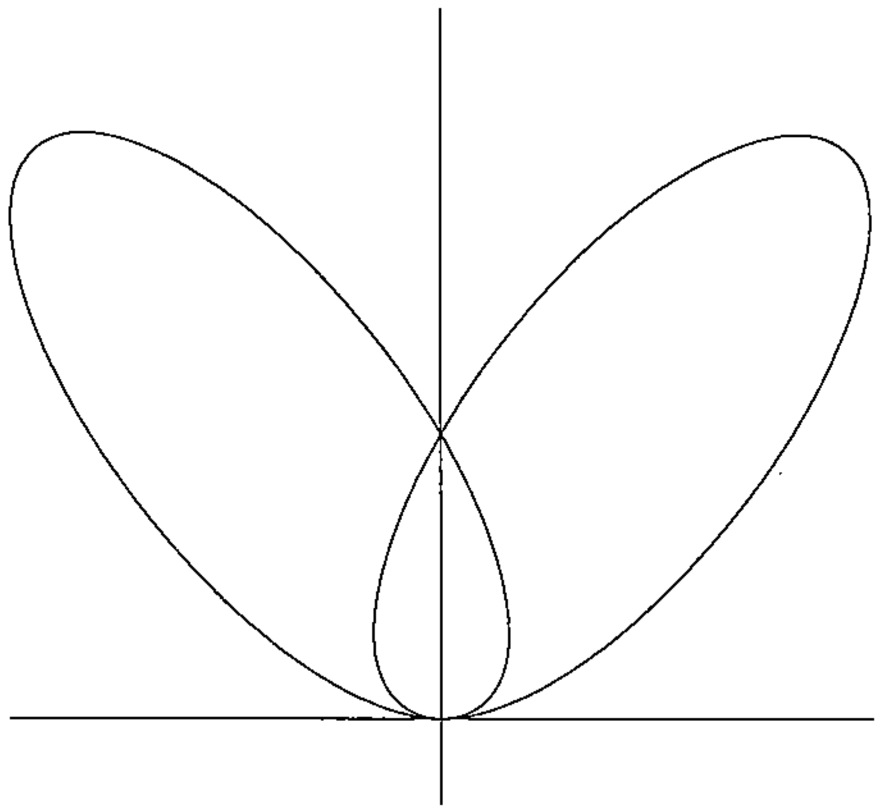


Figure 3.7
Two Real Components Touching
$y^2 - x^4 - y^4 = 0$

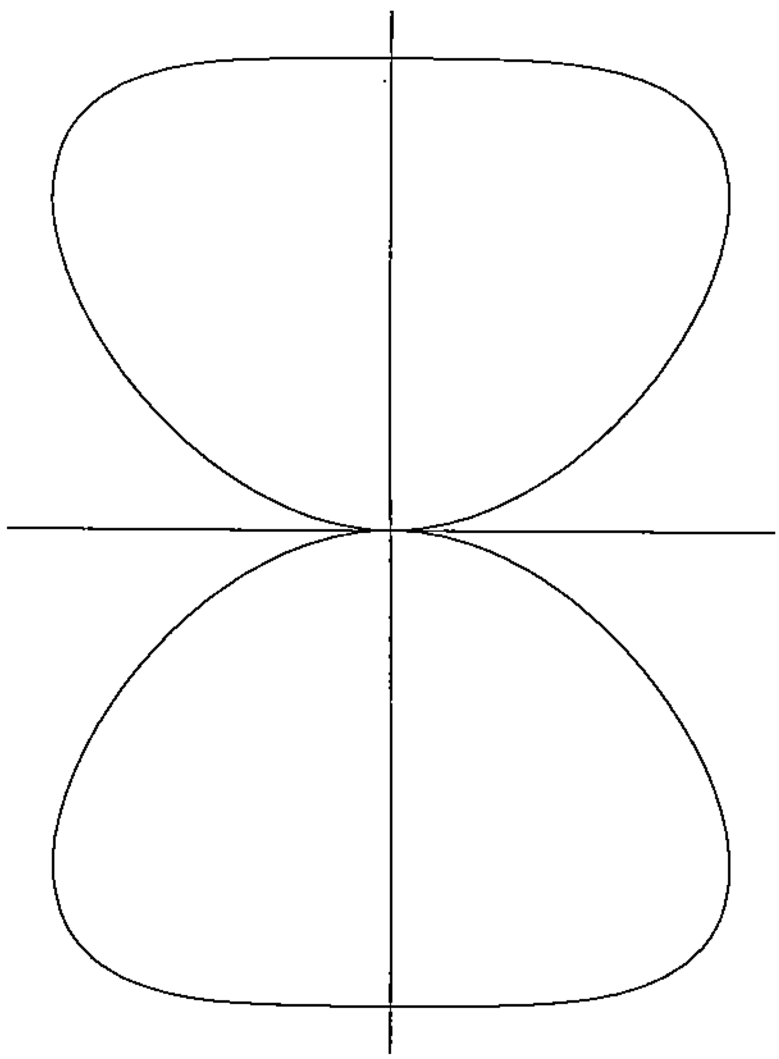
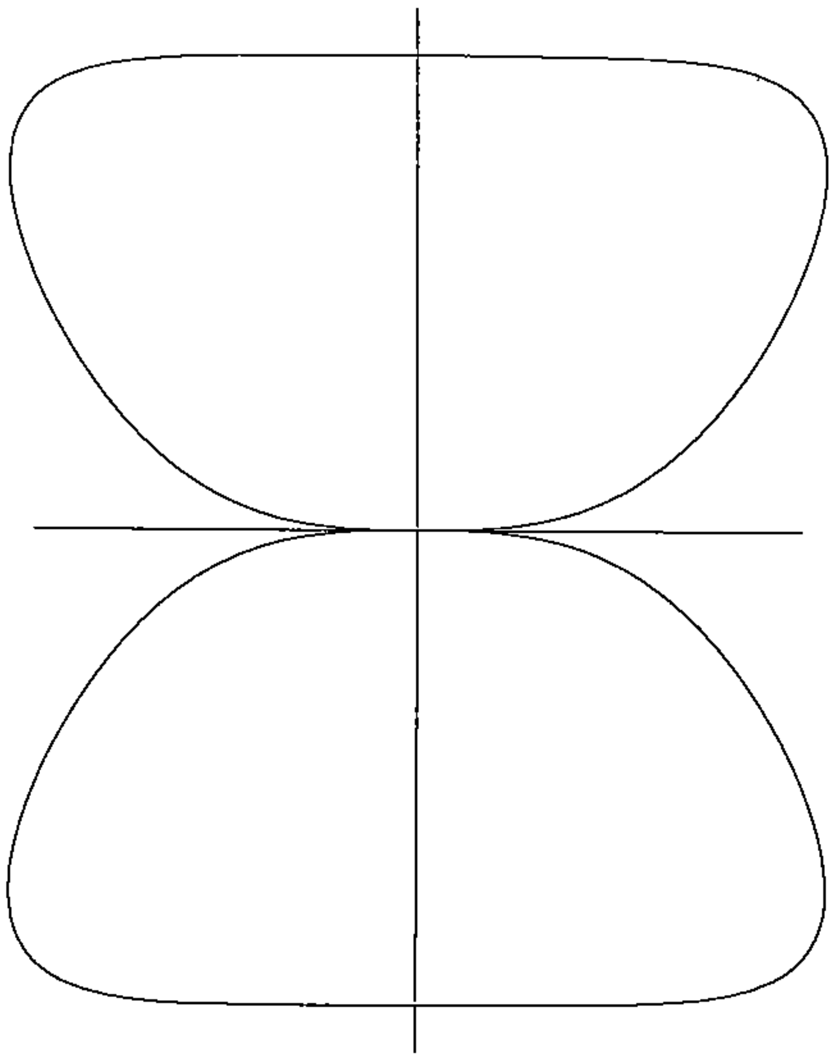


Figure 3.8
One Self-Intersecting Real Component
$y^2 - x^6 - y^6 = 0$



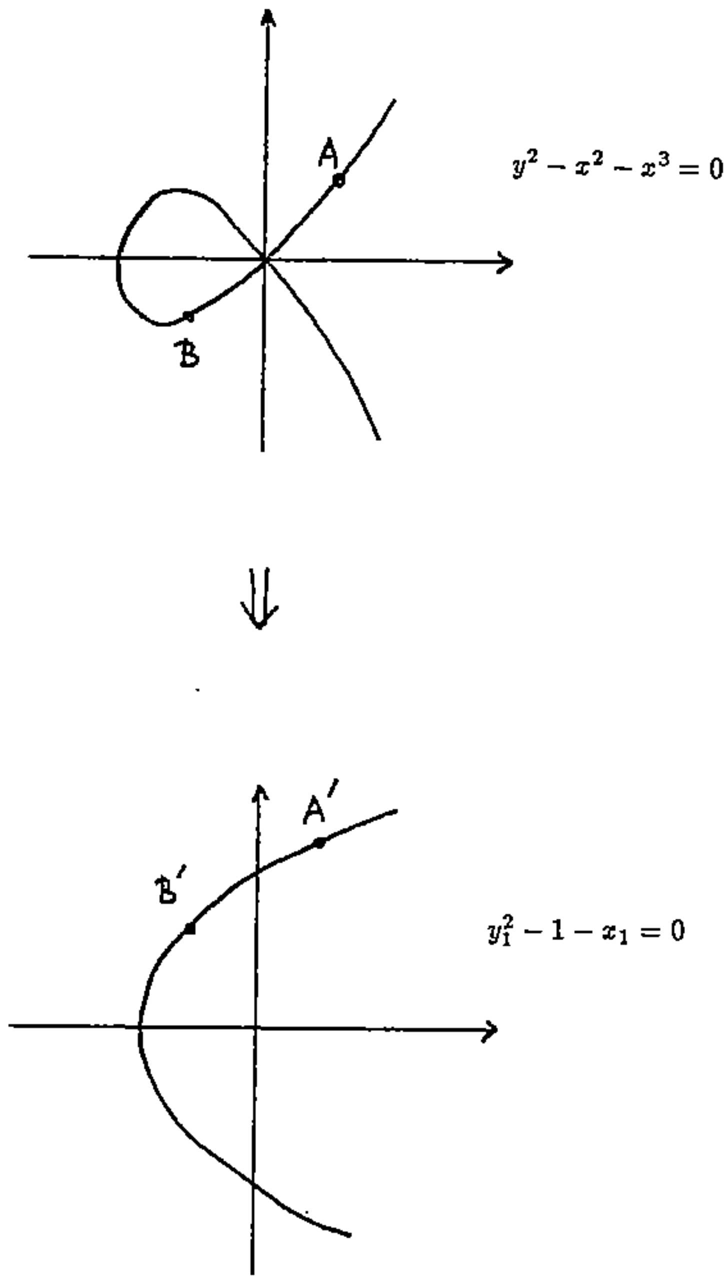


Figure 5.1
Desingularization of a Nodal Singularity

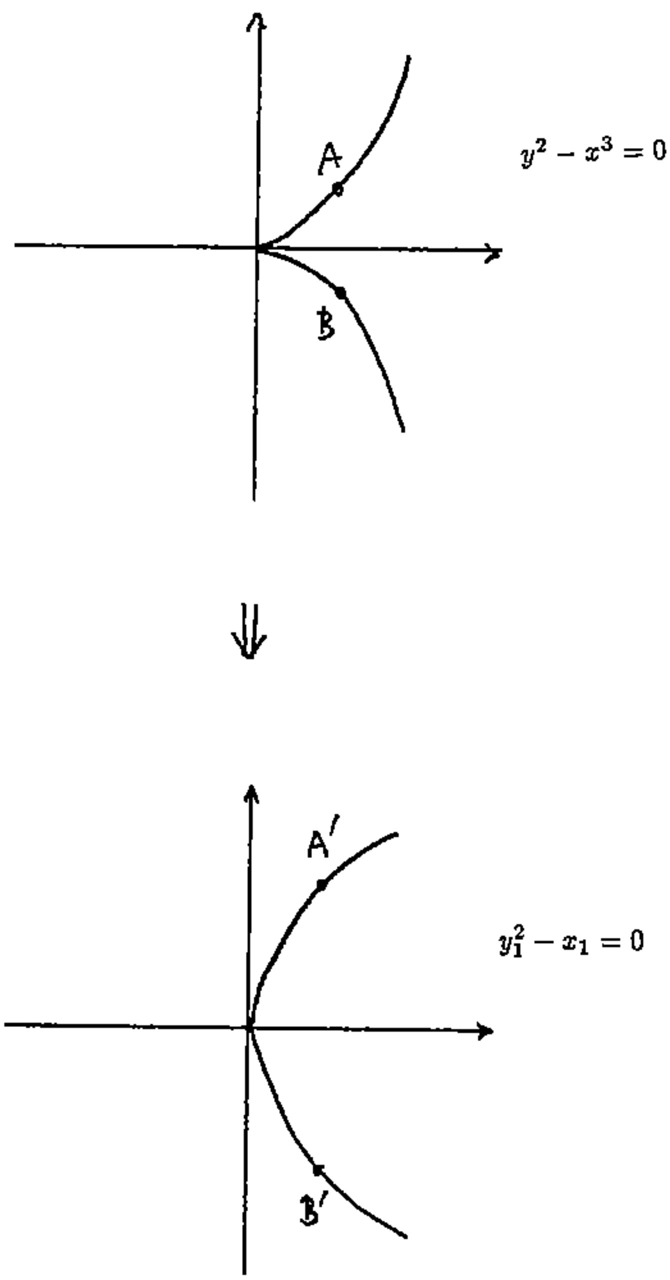
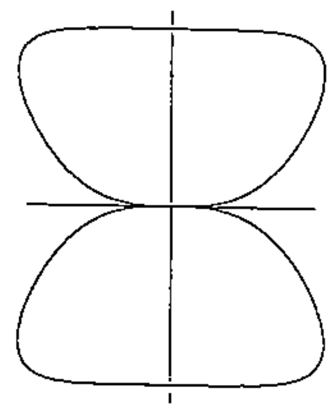


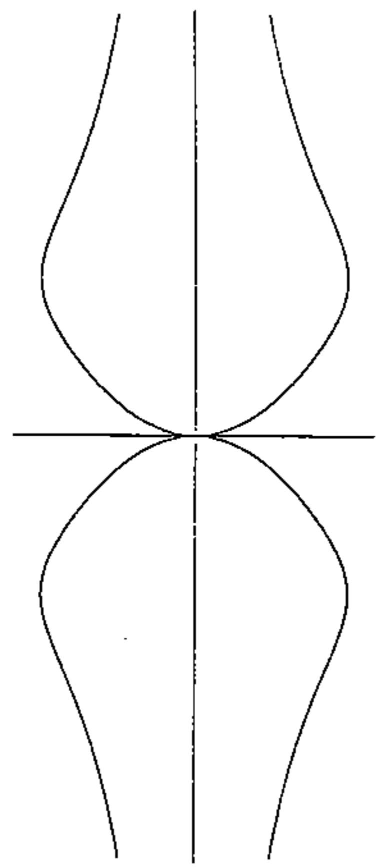
Figure 5.2
Desingularization of a Cuspidal Singularity

Desingularization of $y^2 - x^6 - y^6 = 0$

$$y^2 - x^6 - y^6 = 0$$



$$y_1^2 - x_1^4 - y_1^6\, x_1^4 = 0$$



$$y_2^2 - x_2^2 - y_2^6\, x_2^8 = 0$$

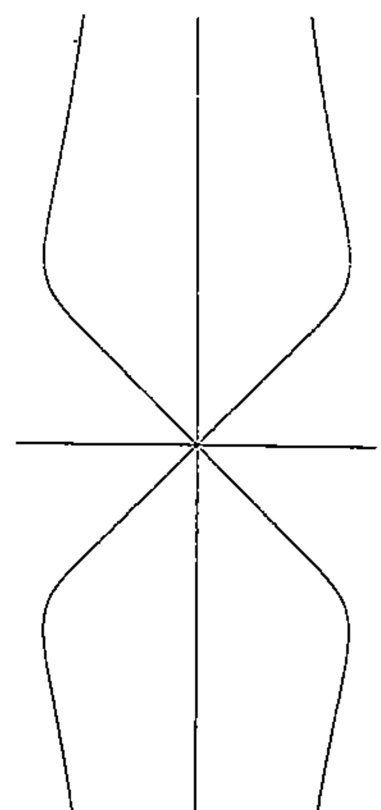


Figure 5.3
Recursive Desingularization

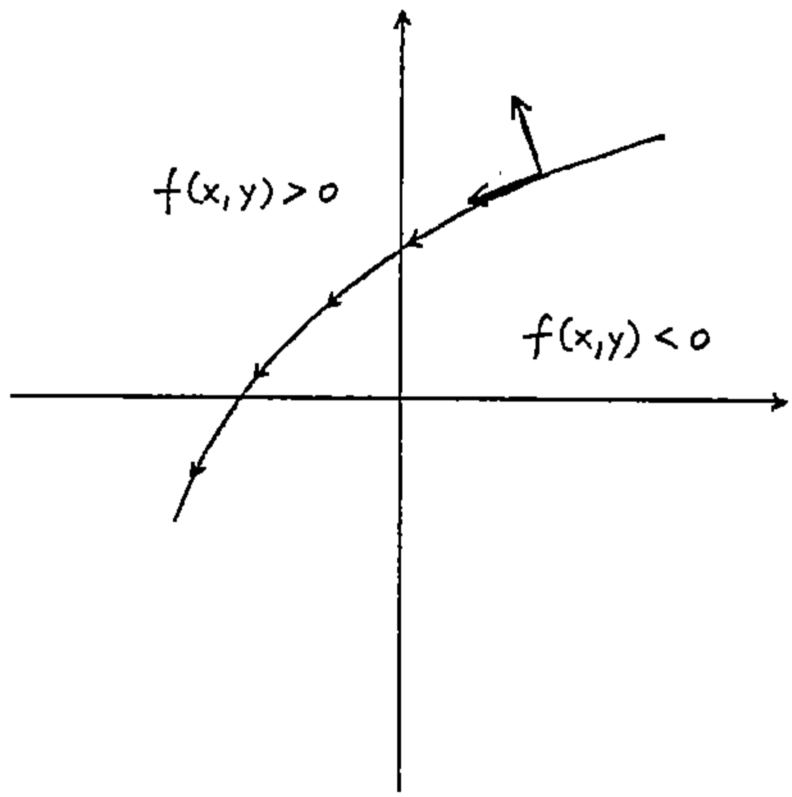


Figure 5.4
Standard Curve Orientation

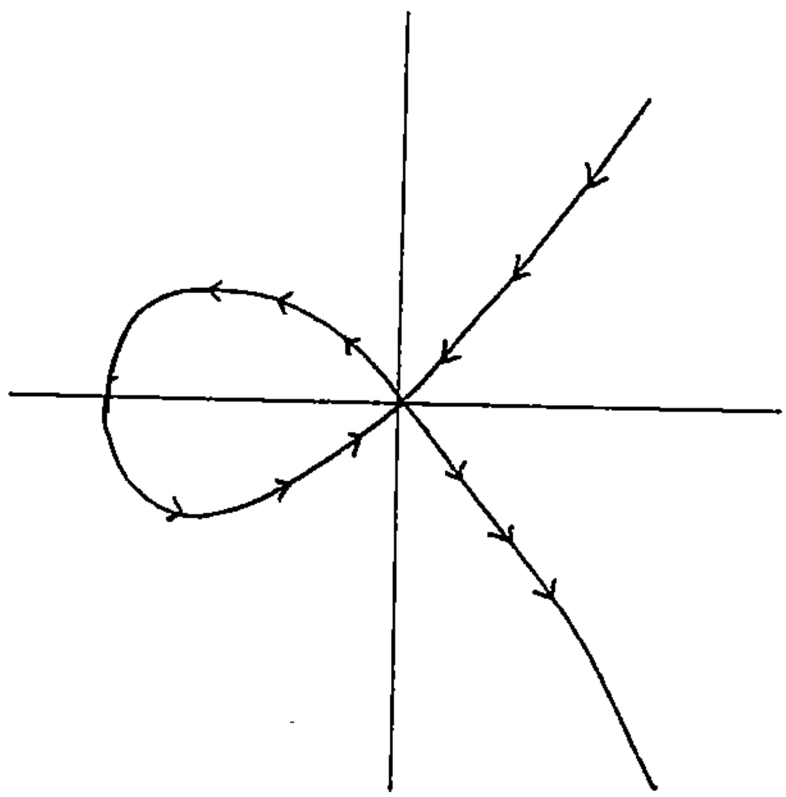


Figure 5.5
Orientation Reversal at a Singularity
$y^2 - x^2 - x^3 = 0$

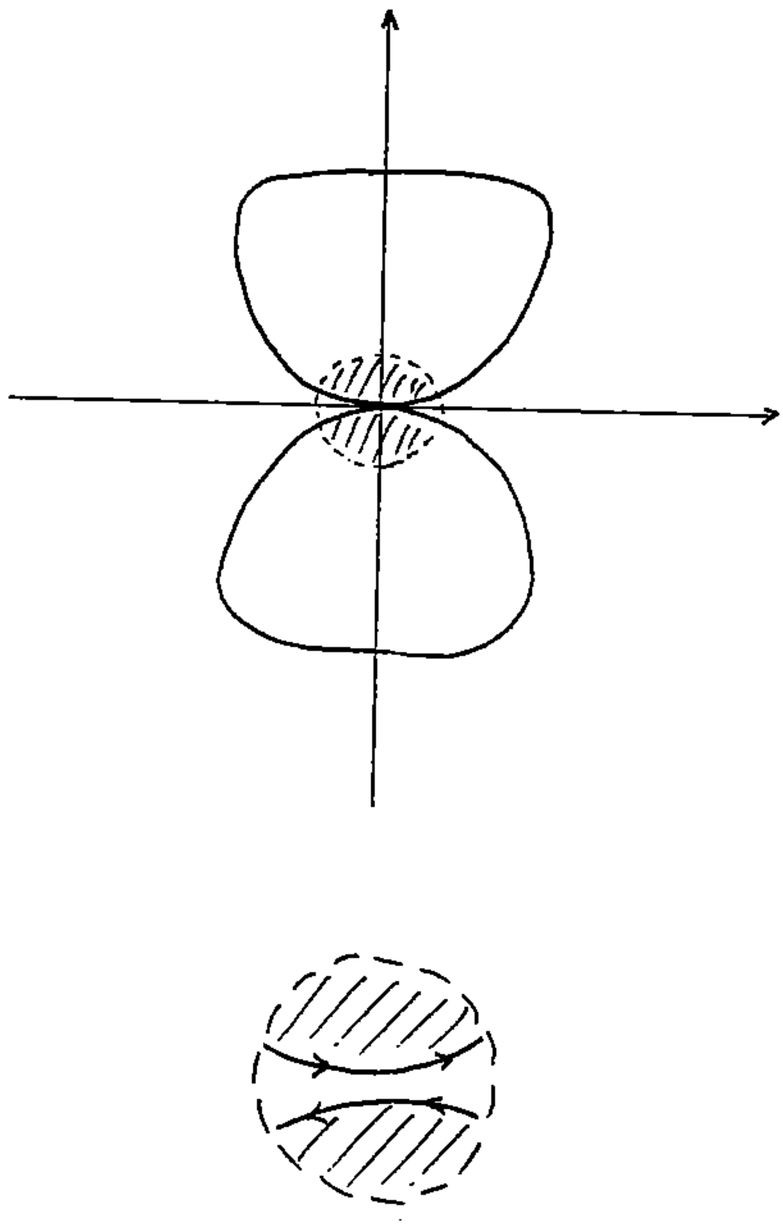


Figure 5.6
Schematic of Curve Topology at the Singularity
Noncrossing Branches
$y^2 - x^4 - y^4 = 0$

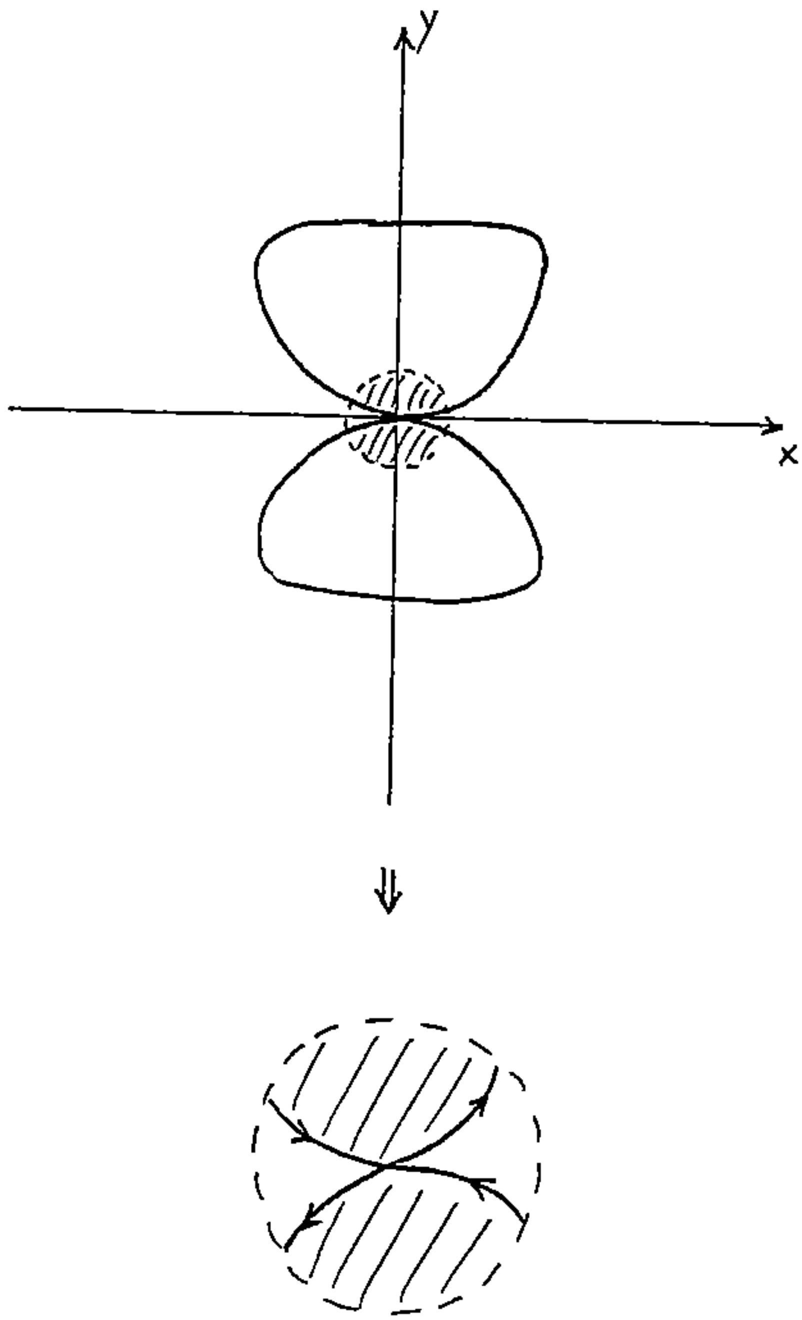


Figure 5.7
Schematic of Curve Topology at the Singularity
Crossing Branches
$y^2 - x^6 - y^6 = 0$

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

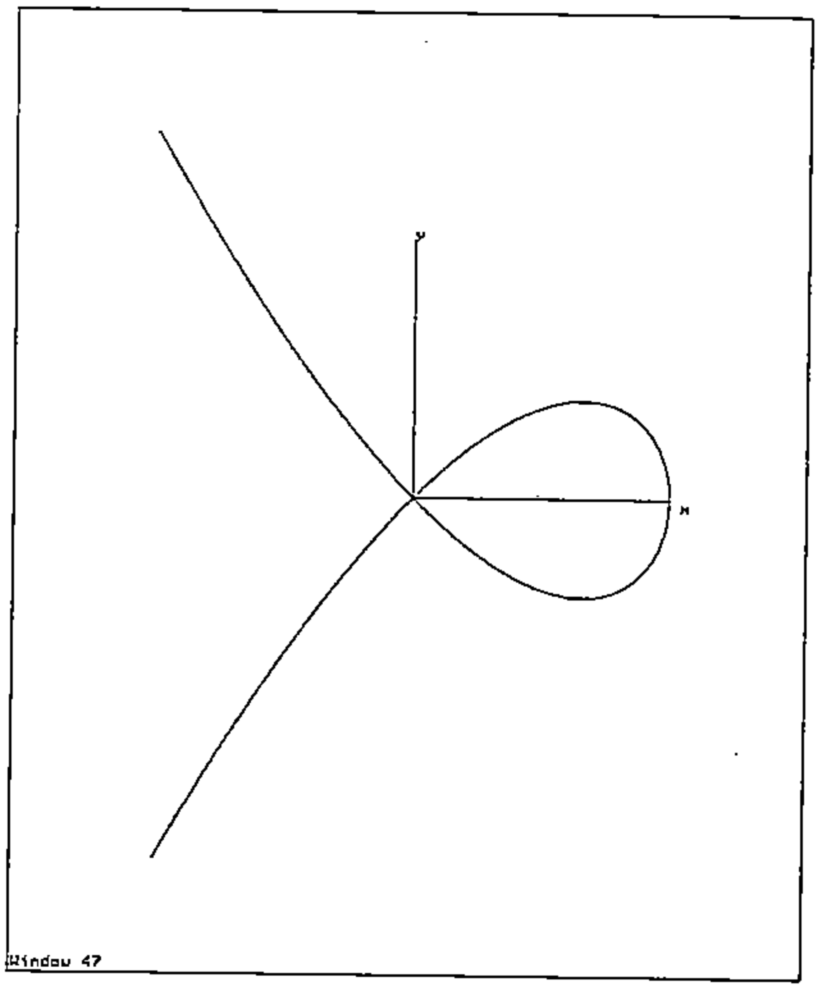


Figure 5.8
$y^2 - x^2 + x^3 = 0$

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

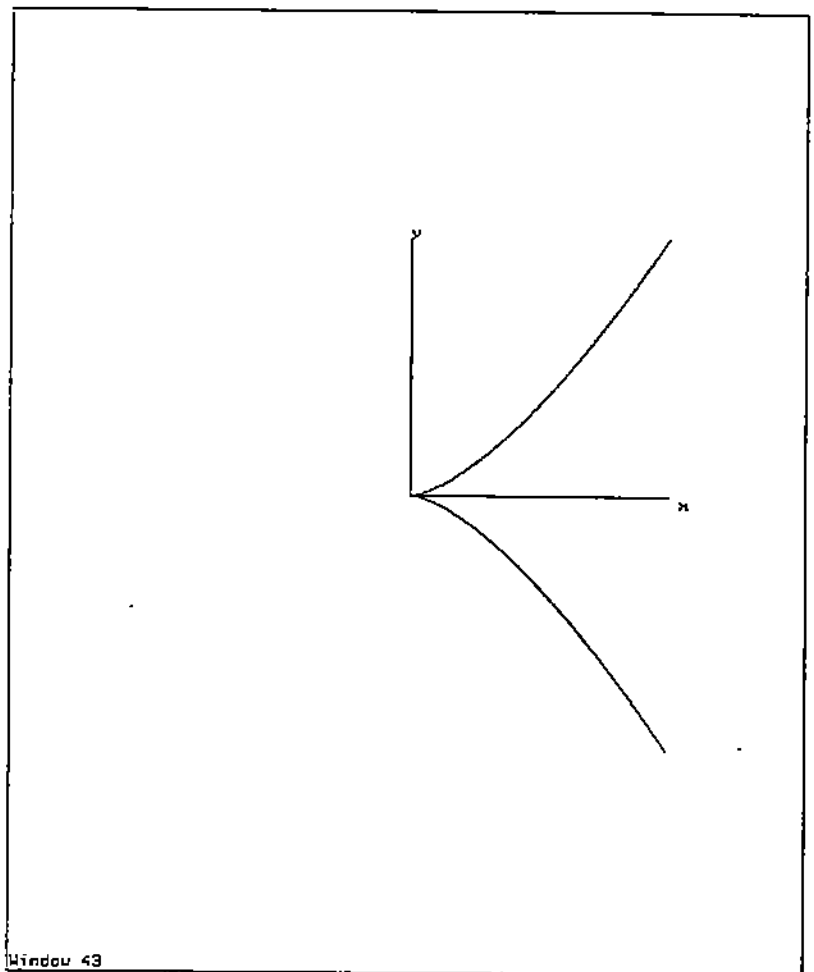


Figure 5.9
$y^2 - x^3 = 0$

$$x^6 - x^2y^3 - y^5 = 0$$

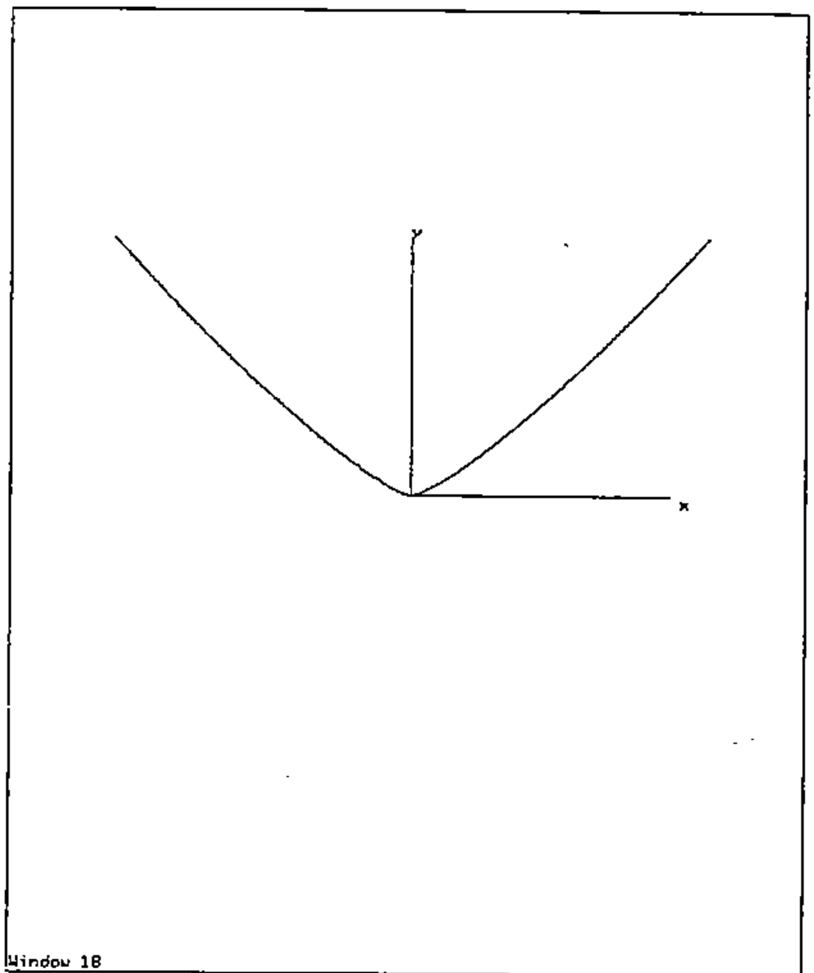


Figure 5.10
$x^6 - x^2y^3 - y^5 = 0$

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

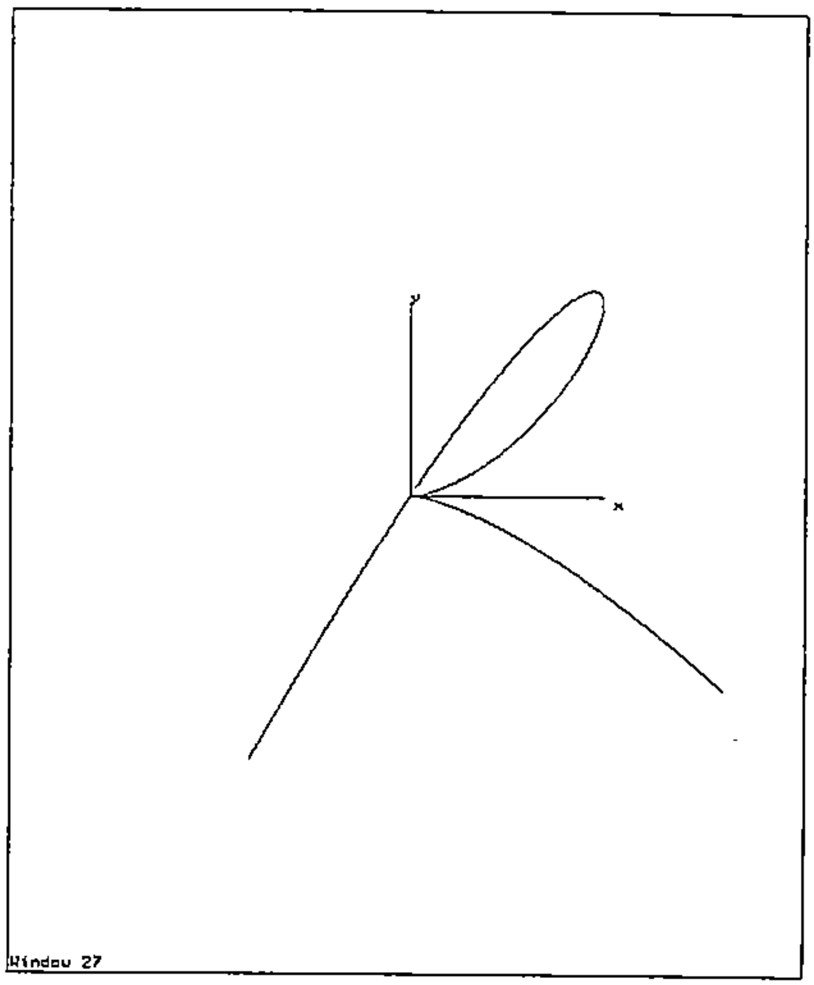


Figure 5.11
$x^4 - 3xy^2 + 2y^3 = 0$

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

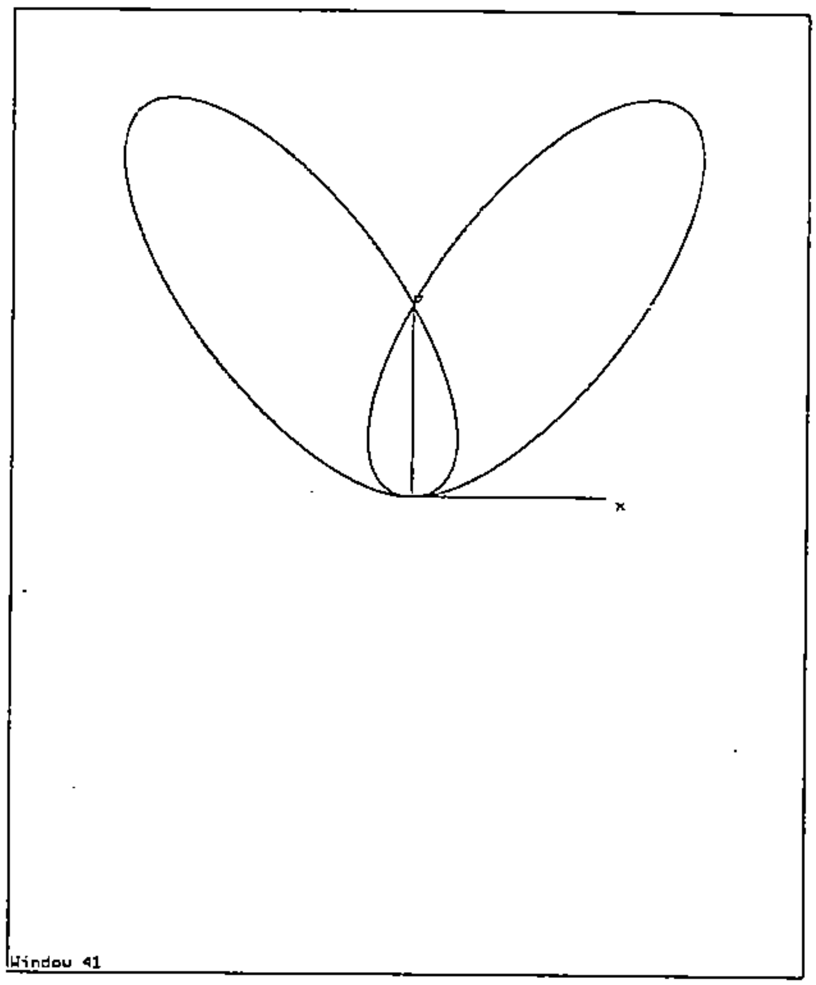


Figure 5.12
$2x^4 - 3x^2y + y^2 - 2y^3 + y^4 = 0$

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

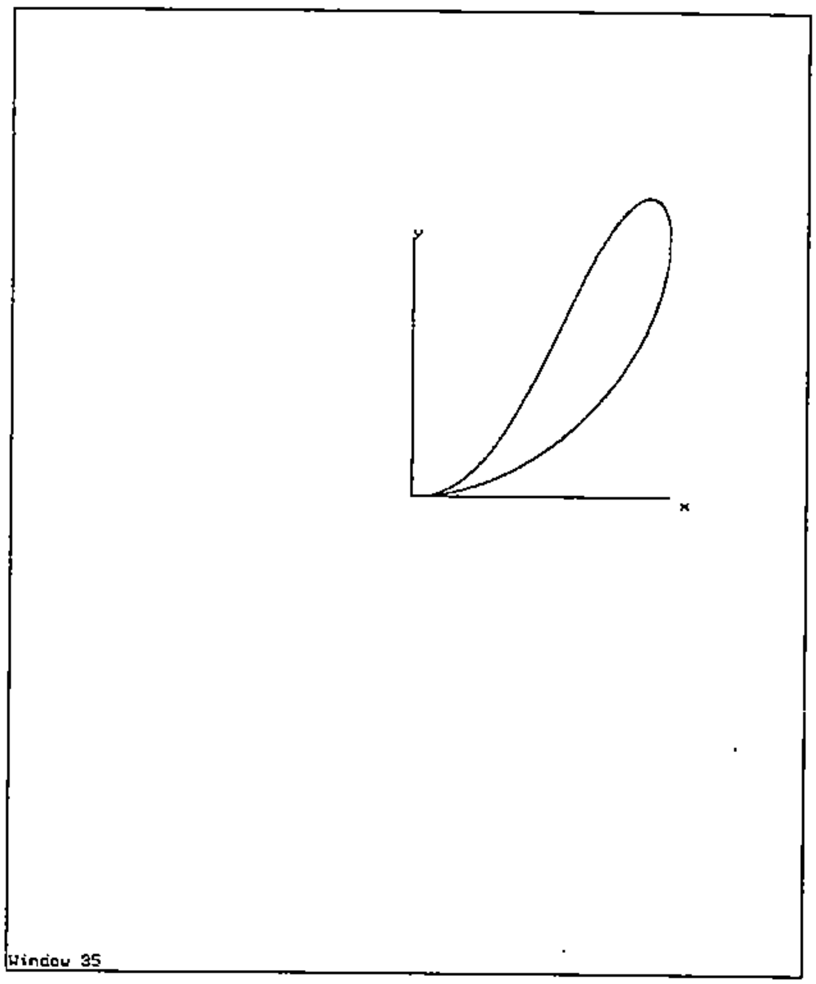


Figure 5.13
$x^4 + x^2y^2 - 2x^2y - xy^2 + y^2 = 0$

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

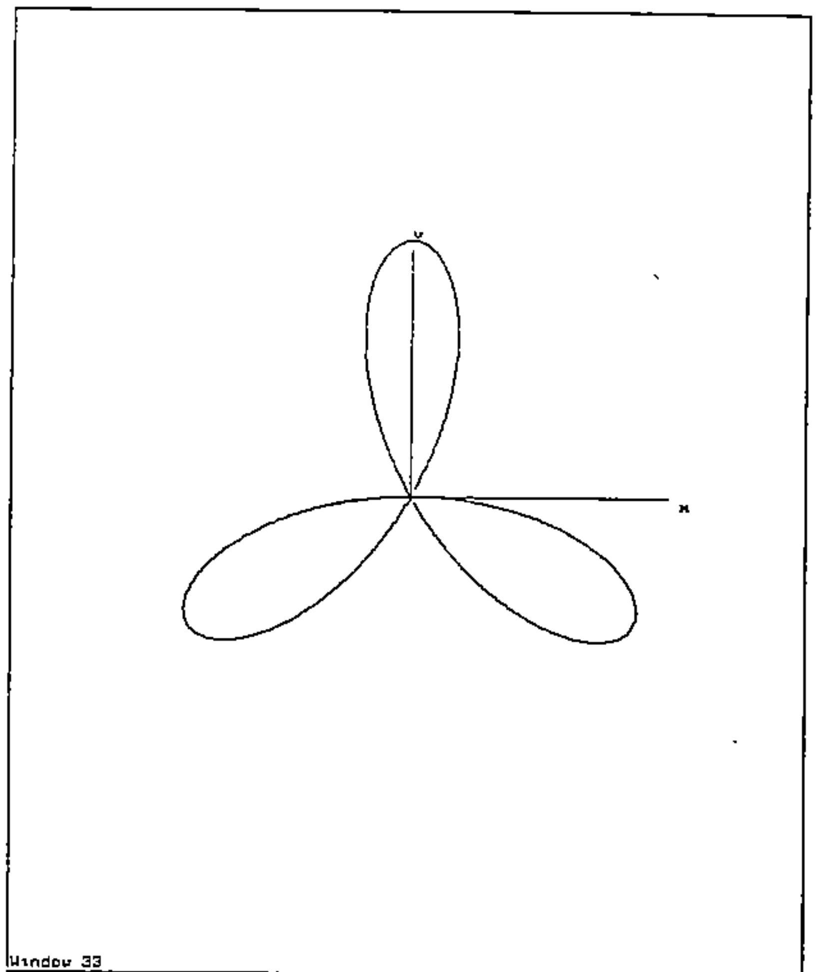


Figure 5.14
$(x^2 + y^2)^2 + 3x^2y - y^3 = 0$

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

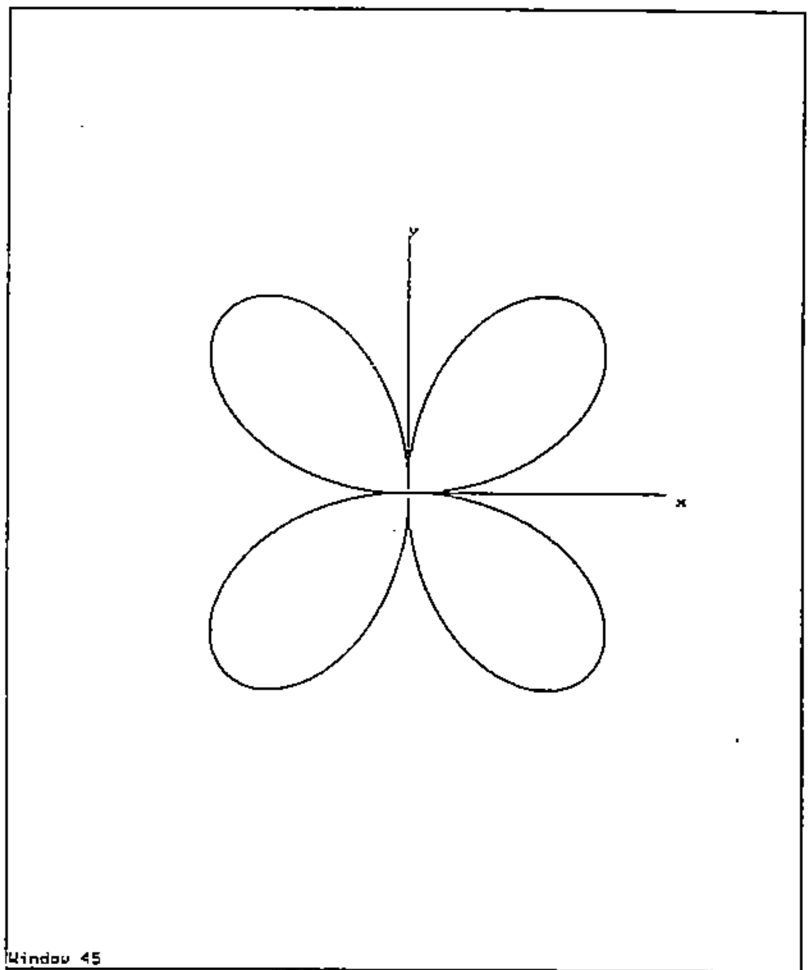


Figure 5.15
$(x^2 + y^2)^3 - 4x^2y^2 = 0$