LOCAL PARAMETERIZATION, IMPLICITIZATION AND INVERSION OF REAL ALGEBRAIC CURVES

(Extended Abstract)

Chanderjit Bajaj*

Computer Sciences Department
Purdue University

Technical Report CSD-TR-863
CAPO Report CER-89-9
February, 1989

* Supported in part by ARO Contract DAAG29-85-C0018 and ONR contract N00014-88-K-00402.

Local Parameterization, Implicitization and Inversion of Real Algebraic Curves

*Chanderjit L. Bajaj**
Computer Science Department,
Purdue University

1 Introduction

Preliminaries

An algebraic plane curve of degree n is implicitly defined by a single polynomial equation $f(x,y) = 0$ of degree n. A rational algebraic curve of degree n can additionally be defined by rational parametric equations which are given as $(x = G_1(u), y = G_2(u))$, where G_1 and G_2 are rational functions in u of degree n, i.e., each is a quotient of polynomials in u of maximum degree n. An algebraic space curve, defined by the intersection of two algebraic surfaces[1] can be given either as a pair of polynomial equations $(f_1(x,y,z) = 0$ and $f_2(x,y,z) = 0)$ or as two sets of parametric equations $(x = G_{1,1}(u_1,v_1), y = G_{2,1}(u_1,v_1), z = G_{3,1}(u_1,v_1))$ 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 rational functions. Rational algebraic space curves are additionally representable as $(x = G_1(u), y = G_2(u), z = G_3(u))$, where G_1, G_2 and G_3 are rational functions in u.

Rational curves are only a subset of implicit algebraic curves of the same degree. While all degree two curves (conics) are rational, only a subset of degree three (cubics) and higher degree curves are rational. In general, a necessary and sufficient condition for the global rationality of an algebraic curve of arbitrary degree is given by the Cayley-Riemann criterion: a curve is rational if and only if $g = 0$, where g, the genus of the curve is a measure of the deficiency of the curve's singularities from its maximum allowable limit [18].

*Supported in part by NSF grant DMS 8816286, ARO Contract DAAG29-85-C0018 and ONR contract N00014-88-K-0402

[1]In modeling the boundary of physical objects it suffices to consider only space curves defined by the intersection of two surfaces. Space curves in general can be defined by several surfaces, however this representation is difficult to handle equationally. General space curves is a topic with various unresolved issues of mathematical and computational interest and an area of future research.

The Problem

Here we wish to consider all algebraic curves, and specifically of genus higher than zero. For all these curves we wish to compute rational parameterizations in the local neighborhood of a point on the curve. This is always possible. At simple points of the curve a straighforward Taylor series expansion, followed by a truncation or a rational Padé approximation, proves sufficient. At a curve's singular point the problem is slighlty more complex, as the Taylor series is not defined. Nevertheless, the curve can be factored into a finite number of power series at the singular point, and rational approximations can be constructed from those.

In particular then our problems are:

- For an implicilty defined algebraic plane curve, compute an approximate rational parametric representation $(x = H_1(t), y = H_2(t))$, for each real branch incident at a point $p = (a_0, b_0)$ on the curve, where $H_1(t)$, $H_2(t)$ are rational functions over the Reals.

- For a parameterically defined algebraic plane curve, compute an approximate implicit representation $f(x, y) = 0$, and an inverse relation $t = F(x, y)$ valid about a point $p = (a_0, b_0)$ on the curve, where $f(x, y)$ is a polynomial and $F(x, y)$ s a rational function over the Reals.

- For an implicitly defined algebraic space curve, compute an approximate rational parametric representation $(x = H_1(t), y = H_2(t), z = H_3(t))$, for each real branch incident at a point $p = (a_0, b_0, c_0)$ on the curve, where $H_1(t)$, $H_2(t)$, $H_3(t)$ are rational functions over the Reals.

- For a parameterically defined algebraic space curve, compute an approximate implicit representation $(f_1(x, y, z) = 0, f_2(x, y, z) = 0)$, and an inverse relation $t = F(x, y, z)$ valid about a point $p = (a_0, b_0, c_0)$ on the curve, where $(f_1(x, y, z) = 0, f_2(x, y, z) = 0)$ is a polynomial and $F(x, y, z)$ is a rational function over the Reals.

Applications:

Rationality of the algebraic curve or surface is a restriction where advantages are obtained from having both the implicit and rational parametric representations [5], [15]. While the rational parametric form of representing a curve or surface allows greater ease for transformation and shape control, the implicit form is preferred for testing whether a point is on the given curve or surface and is further conducive to the direct application of algebraic techniques. Simpler algorithms are possible when both representations are available. For example, a straightforward method exists for

computing curve - curve and surface - surface intersections when one of the curves, respectively surfaces, is in its implicit form and the other in its parametric form. Global parameterization algorithms for plane curves of genus zero, are presented in [1].

There are also numerous applications where explicit local parameterizations, implicitizations, and inversion formulas, which we present here, prove useful in an essential way:

1. Determining the topological type of a real algebraic curve, see for e.g. [3, 11].
2. Adaptive stepping, for curve tracing through singularities, see for e.g [6].
3. Local intersection representation, see for e.g. [14].
4. Piecewise rational approximation for non-rational algebraic curves, i.e., curves of positive genus, see for e.g. [16, 17].

Prior Work

In [6, 14], power series are constructed to locally approximate plane algebraic curves and surface intersections. The method of [14] technically relies on the Implicit Function Theorem, seeking to represent a curve branch explicitly in one coordinate as function of the other coordinate(s), while [6] uses a Taylor series expansion. Both these methods however do not seem to have a natural extension that handles singular points. Further, [16, 17] also present techniques for curve approximation which work only for special cases.

Methods for computing local branch parameterizations at singular points have been presented in [10, 11, 12], both based on the Newton polygon, see for e.g., [18]. We instead use the iterative lifting technique of Hensel together with the fast univariate Padé algorithm of [7]. Local implicitization is considered in [9] extending the technique of [14] of reducing it to solving a linear system of equations. Our techniques are much more direct, requiring only the efficient power series composition and reversion of [8, 13] and straightforward rational function simplification.

Results:

In this paper we present a combination of both algebraic and numerical techniques to achieve local parameterizations about singular points of algebraic curves. We show how to obtain real Weierstrass and Newton power series factorizations using the technique of Hensel lifting. These, together with rational Padé approximations, are used to efficiently construct locally approximate,

rational parametric representations for all real branches of an algebraic plane curve about its singularities. Next we use power series composition and reversion techniques together with rational Padé approximations to efficiently construct locally approximate implicit and inverse representations for parametric algebraic plane curves. Extensions are then given to construct locally approximate, rational parameterizations, implicitizations and inversions for branches of surface intersection space curves. Implementations of these methods and our experiences with them are also discussed.

2 Power Series Computations

2.1 Hensel Lifting

Consider $f(x,y)$ of degree n. Assume it is monic in y . Otherwise, factor out the largest common power of x amongst the terms of f.

$$f(x,y) = f_0(y) + f_1(y)x + \cdots + f_k(y)x^k + \cdots$$

We wish to compute real power series factors $g(x,y)$ and $h(x,y)$ where $f(x,y) = g(x,y)h(x,y)$. The technique of Hensel lifting allows one to reconstruct the power series factors

$$\begin{aligned} g(x,y) &= g_0(y) + g_1(y)x + \cdots + g_i(y)x^i + \cdots \\ h(x,y) &= h_0(y) + h_1(y)x + \cdots + h_j(y)x^j + \cdots \end{aligned} \tag{1}$$

from initial factors $f(0,y) = f_0(y) = g_0(y)h_0(y)$.

Consider the factorization of $f(0,y) = f_0(y)$ as the base case of $k = 0$. Assume $f_0(y)$ is of degree n. Choose real coprime factors $g_0(y)$ of degree p and $h_0(y)$ of degree q satisfying: $p + q = n$. Real coprimeness is achieved by ensuring that g_0 and h_0 contain distinct real roots of f_0 and that complex conjugate pairs are not split up. For the case $n = 2$ however, it may arise that the only coprime factors of f_0 are complex, i.e., the distinct roots are complex conjugates. In that case there only exist complex power series solutions. Since $GCD(g_0(y), h_0(y)) = 1$ using the fast GCD algorithm we can also compute $\alpha(y)$ and $\beta(y)$ such that $\alpha(y)g_0(y) + \beta(y)h_0(y) = 1$

In the iterative Case of $k \geq 1$, we compute $g_k(y)$ and $h_k(y)$ of the desired factorization (1), with degree of $g_k(y) < p$ and degree of $h_k(y) < q$, as follows. We note from (1) that

$$f_k(y) = \sum_{i+j=k} g_i(y)h_j(y)$$

and additionally

$$f_k(y) - \sum_{i < k \wedge j < k} g_i(y)h_j(y) = g_0(y)h_k^*(y) + h_0(y)g_k^*(y) \quad (2)$$

Hence,

$$h_k^*(y) = \alpha(y)[f_k(y) - \sum_{i < k \wedge j < k} g_i(y)h_j(y)]$$
$$g_k^*(y) = \beta(y)[f_k(y) - \sum_{i < k \wedge j < k} g_i(y)h_j(y)]$$

If degree $h_k^*(y) \geq q$ then compute $h_k(y) = h_k^*(y) \bmod h_0(y)$ and set $g_k(y) = \gamma(y)g_0(y) + g_k^*(y)$ where $h_k^*(y) = \gamma(y)h_0(y) + h_k(y)$.

$$f_k(y) - \sum_{i < k \wedge j < k} g_i(y)h_j(y) = g_0(y)h_k(y) + h_0(y)g_k(y) \quad (3)$$

Clearly degree $h_k(y)$ is $< q$. Additionally in (3) the degree of $g_k(y)$ must also be $< p$. This is so because in (3) the degree of the LHS is $< n$ and since degree $g_0(y)h_k(y)$ is $< n$ and degree $h_0(y)$ is $= q$, it must be that degree $g_k(y)$ is $< p$.

Similarly if degree $g_k^*(y) \geq p$ then compute $g_k(y) = g_k^*(y) \bmod g_0(y)$ and set $h_k(y) = \delta(y)h_0(y) + h_k^*(y)$ where $g_k^*(y) = \delta(y)g_0(y) + g_k(y)$. Again, from similar degree arguments as above, is easily seen that the degree bounds of $h_k(y)$ and $g_k(y)$ are met.

2.2 Weierstrass Factorization

Consider $f(x,y)$ with degree n and $ord_y\ f(0,y) = d < \infty$. An $ord_y\ f(0,y) = \infty$ corresponds to $f(0,y) = 0$. This can easily be rectifid by a simple linear transformation of $f(x,y)$, which yields a nonzero $f(0,y)$ and hence a finite $ord_y\ f(0,y)$. We wish to compute a power series factorization of the form $f(x,y) = g(x,y)\underbrace{(y^d + a_{d-1}(x)y^{d-1} + \cdots + a_0(x))}_{h(x,y)}$ where $g(x,y)$ is a unit power series, i.e., $g(0,0) \neq 0$ while $h(x,y)$ is a polynomial in y with coefficients $a_i(x)$, $i = 0 \ldots n-1$ being non-unit power series, i.e., $a_i(0) = 0$. Such a factorization is known as a Weierstrass preparation and is always possible as we now show.

The Weierstrass preparation can efficiently be achieved via Hensel Lifting. Given

$$f(x,y) = f_0(y) + f_1(y)x + \cdots + f_k(y)x^k + \cdots$$

with

$$f(0,y) = f_0(y) = \underbrace{(a_0 + a_1 y + \cdots)}_{g_0(y)}\underbrace{y^d}_{h_0(y)}$$

in general for $k \geq 1$, we wish to compute $h_k(y)$ and $g_k(y)$ using Hensel, yielding factors similar to to (1) such that

$$f_k(y) - \sum_{i < k \wedge j < k} g_i(y)h_j(y) = g_0(y)h_k(y) + y^d g_k(y)$$

with degree $h_k(y) < d$.

To achieve this we compute $A(y) = \frac{f_k(y) - \sum_{i < k \wedge j < k} g_i(y)h_j(y)}{g_0(y)}$ and then set $h_k(y) =$ Terms of $A(y)$ with degree $< d$ and $g_k(y) =$ Terms of $A(y)$ with degree $\geq d$.

2.3 Newton Factorization

Consider $f(x, y)$, a monic polynomial in y of degree n, with coefficients polynomial or power series or meromorphic series in x

$$f(x, y) = y^n + a_{n-1}(x)y^{n-1} + \cdots + a_0(x)$$

Then it is possible to factor $f(x, y)$ into linear factors

$$f(x, y) = \Pi_{i=1}^{n} (y - \eta_i(t))$$

with $x = t^m$ and m a positive integer and $\eta_i(t)$ power series or meromorphic series. This factorization can also be achieved via Hensel lifting. We precondition the curve so that it admits a non-trivial base factorization, i.e. having at least two coprime factors which can be lifted.

Step 1: Make $a_{n-1}(x) = 0$ via substitution $\bar{y} = y + \frac{a_{n-1}(x)}{n}$

Step 2: Ensure some $a_{n-i}(0) \neq 0$ for $i \geq 2$ via substitution $\check{y} = \frac{\bar{y}}{x^\lambda}$ with $\lambda = min_{(2 \geq i \geq n)} \frac{\alpha_i}{i}$ and $\alpha_i = ord_x a_{n-i}(x)$. Then $f(0, \check{y}) = f_0(\check{y})$ has at least two distinct roots.

Step 3: Now use Hensel lifting to lift the factorization $f_0(\check{y}) = g_0(\check{y})h_0(\check{y})$ to $f(x, \check{y}) = g(x, \check{y})h(x, \check{y})$.

Repeat Steps 1-3 until all factors are linear or all real factors are obtained.

3 Local Parameterization

Consider an implicit plane algebraic curve $f(x, y) = 0$, with a singularity at the origin. (A singularity can be translated to the origin by a straightforward linear transformation). To compute a local parametric approximation of each of the curve's branches incident at the origin, we execute the following steps:

1. Compute a Weierstrass power series factorization of $f(x,y)$ into $f = gh$, where $g((x,y))$ is a unit power series and $h((x))(y)$ is a polynomial in y with coefficients non-unit power series in x. The equation $h = 0$ corresponds to the curve's branches at the origin while the power eries equation $g = 0$ corresponds to the portion of the plane curve away from the origin.

2. Recursively apply the Newton factorization to $h((x))(y)$ till all factors are linear in y or all real factors are obtained. Each of these power series factors represent a local branch parameterization of the type $x = t^k$ and $y = b_i((t))$ where b_i is a power series. The minimum of k and $ord_t(b_i)$, say d, is known as the order of the branch, with $d > 1$ implying a singular branch or "place" of the curve.

3. For each distinct branch power series parameterization $y = b_i((t))$, compute a Padé rational function approximation.

Consider next an algebraic space curve C, defined implicitly by two equations ($f_1(x,y,z) = 0$ and $f_2(x,y,z) = 0$), and having a singularity at the origin. To compute a local parametric approximation of each of the curve's branches incident at the origin, we execute the following steps:

1. Using birational projection techniques of [4], construct a projected plane curve $P : f_3(x,y)$ and an inverse rational map $z = F(x,y)$ from points on P to points on C.

2. Apply the steps 1., 2., and 3., of the plane curve parameterization algorithm above, to P, to compute all branch parameterizations and local rational Padè approximants. Next use the inverse rational map to yield the local parameterizations of all branches of the space curve at the origin.

4 Local Implicitization

Consider a rational parametric plane curve given by ($x = H_1(t)$, $y = H_2(t)$) where H_1 and H_2 are rational functions over the Reals. To compute a local implicit approximation of the curve around the origin, we execute the following steps:

1. Let $\tau^k = x = H_1(t)$ where $k = ord_t(H_1)$ = power of the lowest degree term of the power series expansion of the rational function. (Wlg we assume $ord_t H_1 = k \geq ord_t H_2 = \ell$, for otherwise we can switch the roles of x and y).

2. Compute $\tau = (H_1)^{1/k} = g_1((t))$ = power series of order 1.

3. Next invert the power series equation $\tau = g_1((t))$ to yield $t = g_1^{-1}((\tau))$. This yields $(x = \tau^k,\ y = H_2(g_1^{-1}((\tau))) = g_2((\tau))$, where $ord_\tau g_2 = ord_t H_2 = \ell$.

4. Now if $\ell = 1$ then invert $y = g_2((\tau))$ to yield $\tau = g_2^{-1}((y))$, and construct a suitable Padé rational function approximant $\tau = H_3(y)$. The local implicit approximation is then $x - H_3^k(y) = 0$.

5. When $\ell > 1$ then let m = least common multiple of ℓ and k, and compute $\frac{y^{m/\ell}}{x^{m/k}} = \frac{g_2^{m/l}}{\tau^m}$ $= \kappa = (1 + c_1\tau + \cdots)^{m/\ell} = g_3((\tau))$, a power series of order 1. Note $\frac{m}{l}$ and $\frac{m}{k}$ are both integers. Next, compute the inverse power series, $\tau = g_3^{-1}((\kappa))$, followed by the rational Padé approximant computation to yield $\tau = G(\kappa) = G(y^{m/l}/x^{m/k})$ where G is a rational function. The local implicit approximation is then the polynomial simplification of the expression $x - G^k(\kappa) = 0$.

Next, consider a rational parametric space curve given by $(x = H_1(t),\ y = H_2(t),\ z = H_3(t))$ where H_1, H_2 and H_3 are rational functions over the Reals. To compute a local implicit approximation of the curve around the origin, we execute the steps 1. to 5. of the above algorithm for the plane curve case, twice. Once for $(x = H_1(t),\ y = H_2(t))$ to yield a local implicit equation $f_1(x, y) = 0$, and then for $(x = H_1(t),\ z = H_3(t))$ to yield a local implicit equation $f_2(x, z) = 0$. Of course steps 1. and 2. are not repeated. The implicit equations $f_1 = 0$ and $f_2 = 0$ are cylinders, containing the space curve C, locally about the origin.

5 Local Inversion

To locally invert a parameterization $(x = H_1(t),\ y = H_2(t))$ about the origin we compute the following:

1. First execute steps 1., 2. and 3. of the last section. Then, as before, let $(x = \tau^k,\ y = H_2(g_1^{-1}((\tau)))$ $= g_2((\tau)))$ represent a branch of the curve through the origin and let $\ell = ord_\tau g_2((\tau))$.

2. Now if $\ell = 1$ then invert $y = g_2((\tau))$ to yield $\tau = g_2^{-1}((y)) = g_3((y))$. Furthermore, $\tau = x g_4^k((y)) = x G_4(y)$, where $g_4((y))$ is the reciprocal power series of g_3, and $G_4(y)$ an appropriate Padé approximant of g_4^k. Now, from step 3. of the last section we know that

$t = g_1^{-1}((\tau))$, from which we construct a suitable Padé rational function approximant $t = G_1(\tau)$. The local inversion formula is then $t = G_1(G_4(x,y)) = G(x,y)$, where G is a rational function.

3. When $\ell > 1$ then let m = least common multiple of ℓ and k, and compute $\frac{y^{m/\ell}}{x^{m/k}} = \frac{g_2^{m/l}}{\tau^m}$ $= \kappa = (1 + c_1\tau + \cdots)^{m/\ell} = g_3((\tau))$, a power series of order 1. Note $\frac{m}{l}$ and $\frac{m}{k}$ are both integers. Next, compute the inverse power series, $\tau = g_3^{-1}((\kappa))$, as well as construct $t = g_1^{-1}(g_3^{-1}((\kappa)))$. This is followed by the rational Padé approximant computation to yield the local inversion formula $t = G(\kappa) = G(y^{m/l}/x^{m/k})$ where G is a rational function.

Next, consider a rational parametric space curve given by $(x = H_1(t),\ y = H_2(t),\ z = H_3(t))$ where H_1, H_2 and H_3 are rational functions over the Reals. To compute a local inversion formula of the curve around the origin, we execute the steps 1. to 3. of the above algorithm for the plane curve case, twice, without repeating any identical substeps. Once for $(x = H_1(t),\ y = H_2(t))$ to yield a local inversion formula $t = G_a(x,y)$, and then for $(x = H_1(t),\ z = H_3(t))$ to yield a local inversion formula $t = G_b(x,y)$. A local inversion formula for the space curve then is $t = \frac{G_a^2}{G_b} = G(x,y)$.

6 Implementation Issues

The algorithms of sections 3, 4, and 5 have been implemented as part of an interactive algebraic geometry package, on a Symbolics Lisp machine using Common Lisp and C. The Hensel power series computations of section 2.1, as well as its use in sections 2.2, and 2.3 are based on a robust implementation of the fast euclidean HGCD algorithm [2, 7]. Rational Padé approximants are also computed based on the same HGCD algorithm, [7]. Power Series are stored as truncated sparse polynomials, as are the original algebraic curves, viz., a list of degree, variable list and term list, with nonzero terms stored as coefficient and exponents. Floating point coefficients are allowed in the input curve representations, which are then converted to rational numbers for the GCD and power series computations. In Newton factorizations, user options are provided to compute only real branch factorizations. This is achieved by not allowing complex conjugate roots of the appropriate univariate polynomial, to split in the base case of the Henselian computation.

Examples from the software implementation, are shown in Figures 1. 2. and 3. at the end of the paper. Figure 1.1 shows an implicitly defined quartic plane curve with a tacnodal singularity at the origin. The corresponding Figure 1.2 shows the local parameterization of the two real branches

at the origin, as well as a (2,3) Padé approximations. Figures 2.1 and 2.2 and Figures 3.1 and 3.2 are other similar examples of quartic and sextic curves.

7 Conclusions and Future Research

The results of this paper are being extended to deal with power series computations in two or more variables. These would yield a faster solution to the branch factorizations and local parameterization of space curves, since the power series expansions of an implicit algebraic surface, about a point of interest, can then be directly substituted into the other implicit surface equation of the implicitly defined space curve. Note, that the methods of section 2. work even if the input equations are power series, as would be the case then.

In particular then, our future goals are to efficiently compute

1. Power series expansions about singular points and curves on surfaces, to yield bivariate local parameterization, implicitization and inversion algorithms.

2. Generate suitable expansion points and curves for a piecewise rational surface approximation.

Acknowledgements: I wish to thank Professor Abhyankar for many useful explanations on singularities and Insung Ihm for all the time he spent on the software implementation.

References

[1] Abhyankar, S., and Bajaj, C., (1988), "Computations with Algebraic Curves", *Proc. of the Intl. Symposium on Symbolic and Algebraic Computation*, ISSAC88, to appear.

[2] Aho, A., Hopcroft, J., and Ullman, J., (1974), "The Design and Analysis of Computer Algorithms", Addison Wesley, Reading MA.

[3] Arnon, D.S., and McCallum, S., (1983), "A Polynomial-time Algorithm for the Topological Type of a Real Algebraic Curve", Tech. Rept. 454, Comp. Science Dept., Purdue University. See also *J. of Symb. Comput.*.

[4] Abhyankar, S., and Bajaj, C., (1989), "Automatic Rational Parameterization of Curves and Surfaces IV: Algebraic Space Curves", *ACM Transactions of Graphics*, to appear.

[5] Bajaj, C., (1988a), "Geometric Modeling with Algebraic Surfaces", *The Mathematics of Surfaces III*, Oxford University Press, to appear.

[6] Bajaj, C., Hoffmann, C., Hopcroft, J. and Lynch, R., (1988), "Tracing Surface Intersections", *Computer Aided Geometric Design*, 5, 285 - 307.

[7] Brent, R., Gustavson, F., and Yun, D., (1980), "Fast Solutions of Toeplitz Systems of Polynomial Equations and Computation of Padé Approximants", *Journal of Algorithms*, 1, 259 - 295.

[8] Brent, R., and Kung, H., (1978), "Fast Algorithms for Manipulating Formal Power Series", *Journal of the ACM*, 25, 4, 581 - 595.

[9] Chuang, J., and Hoffmann, C., (1989) "On Local Implicit Approximation and its Applications", *ACM Transactions on Graphics* , to appear.

[10] Chudnovsky, and Chudnovsky, (1987), "On Expansions of Algebraic Functions in Power and Puiseux Series ", *Journal of Complexity*, part I: 2, 271-294 and part II:3, 1-25.

[11] Cucker, F., Pardo, L., Raimondo, M., Recio, T., and Roy., M., (1989), "On the Computation of the Local and Global Analytic Branches of a Real Algebraic Curve", Preprint, Université de Rennes.

[12] Duval, D., (1987) "Diverses questions relatives au Calcul Formel Avec Des Nombres Algébriques", Doctoral Thesis, ĹUniversité Scientifique, Technologique, Et Medicale De Grenoble.

[13] Kung, H., and Traub (1978), "All Algebraic Functions Can Be Computed fast", *Journal of the ACM*, 25, 2, 245 - 260.

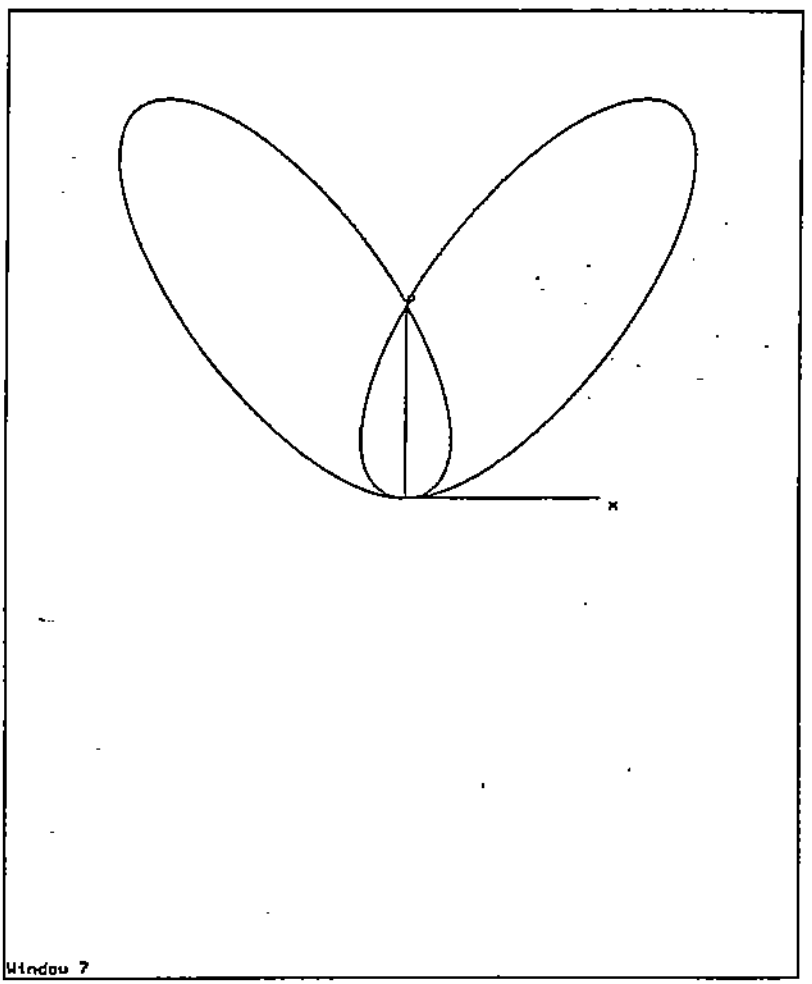
[14] Montaudouin, Y., Tiller, W., and Vold, H., (1986), "Applications of Power Series in Computational Geometry", *Computer Aided Design*, 18, 10, 514 - 524.

[15] Sederberg, T., Anderson, D., Goldman, R., (1985), "Implicitization, Inversion, and Intersection of Planar Rational Cubic Curves", *Computer Vision, Graphics and Image Processing*, 31, 89 - 102.

[16] Sederberg, T., Zhao, J., and Zundel, A., (1988), "Approximate Parameterization of Algebraic Curves" Technical Report ECGL-88104, Brigham Young University.

[17] W. Waggenspack (1987), "Parametric Curve Approximations for Surface Intersections", Ph.D. Dissertation, CADLAB, Purdue University, December 1987.

[18] Walker, R., (1950), *Algebraic Curves*, Springer Verlag 1978, New York



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

Figure 1.1

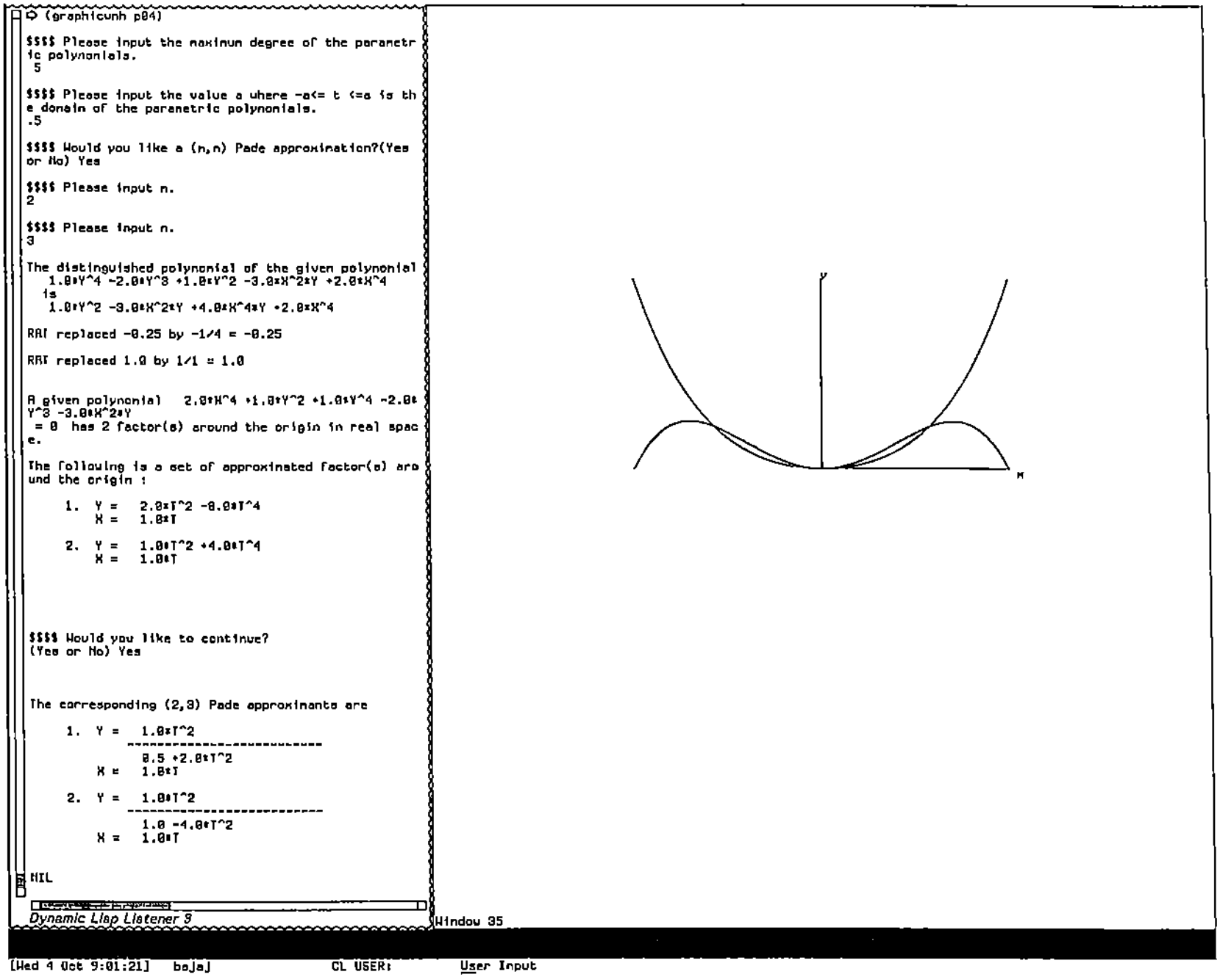
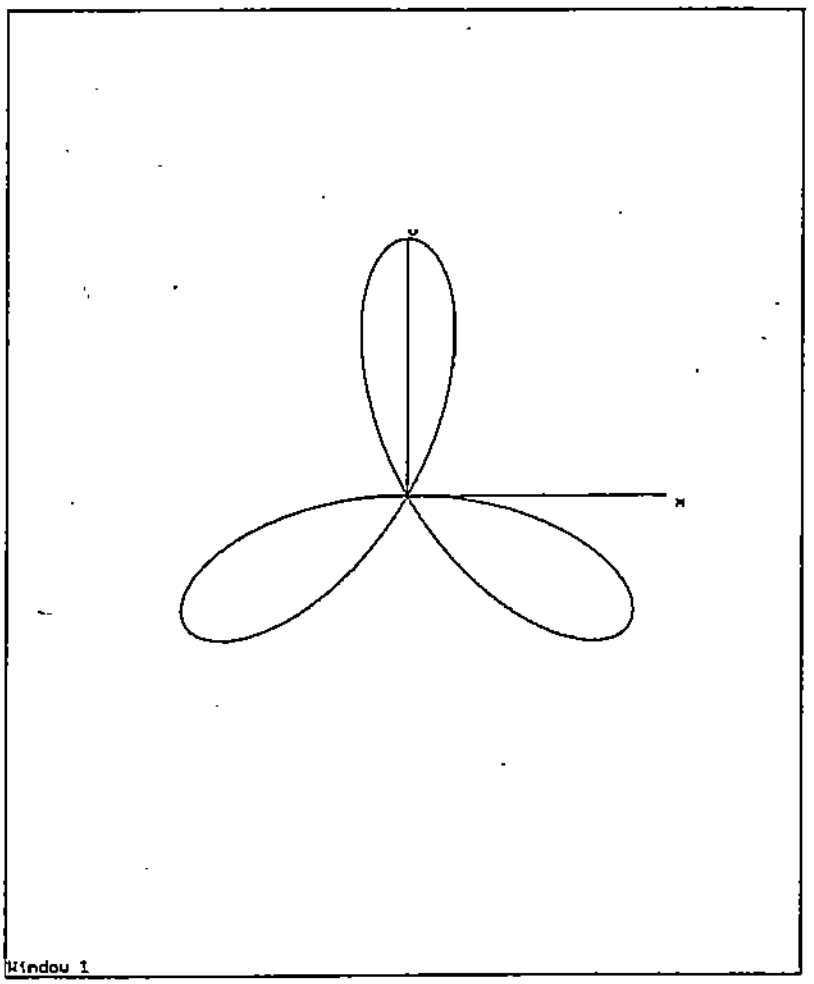

(graphicunh p04)
$$$$ Please input the maximun degree of the parametr
ic polynomials.
5
$$$$ Please input the value a where -a<= t <=a is th
e domain of the parametric polynomials.
.5
$$$$ Would you like a (n,n) Pade approximation?(Yes
or No) Yes
$$$$ Please input n.
2
$$$$ Please input n.
3
The distinguished polynomial of the given polynomial
1.0*Y^4 -2.0*Y^3 +1.0*Y^2 -3.0*X^2*Y +2.0*X^4
is
1.0*Y^2 -3.0*X^2*Y +4.0*X^4*Y +2.0*X^4
RAT replaced -0.25 by -1/4 = -0.25
RAT replaced 1.0 by 1/1 = 1.0
A given polynomial 2.0*X^4 +1.0*Y^2 +1.0*Y^4 -2.0*
Y^3 -3.0*X^2*Y
= 0 has 2 factor(s) around the origin in real spac
e.
The following is a set of approximated factor(s) aro
und the origin :
1. Y = 2.0*T^2 -8.0*T^4
X = 1.0*T
2. Y = 1.0*T^2 +4.0*T^4
X = 1.0*T
$$$$ Would you like to continue?
(Yes or No) Yes
The corresponding (2,3) Pade approximants are
1. Y = 1.0*T^2
8.5 +2.0*T^2
X = 1.0*T
2. Y = 1.0*T^2
1.0 -4.0*T^2
X = 1.0*T
NIL
Dynamic Lisp Listener 9
Window 35
Y
X
[Wed 4 Oct 9:01:21] baja CL USER: User Input

Figure 1.2



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

Figure 2.1

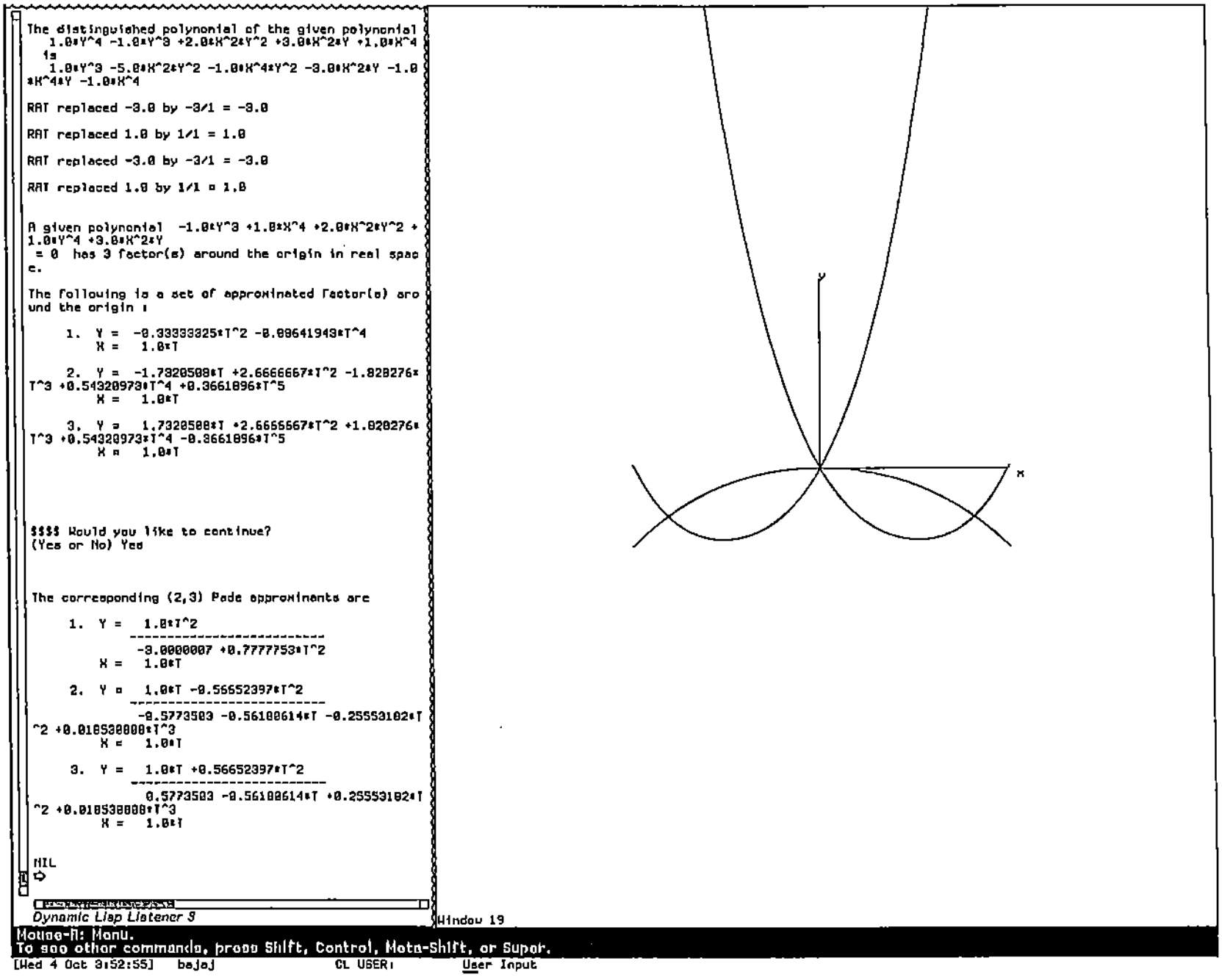
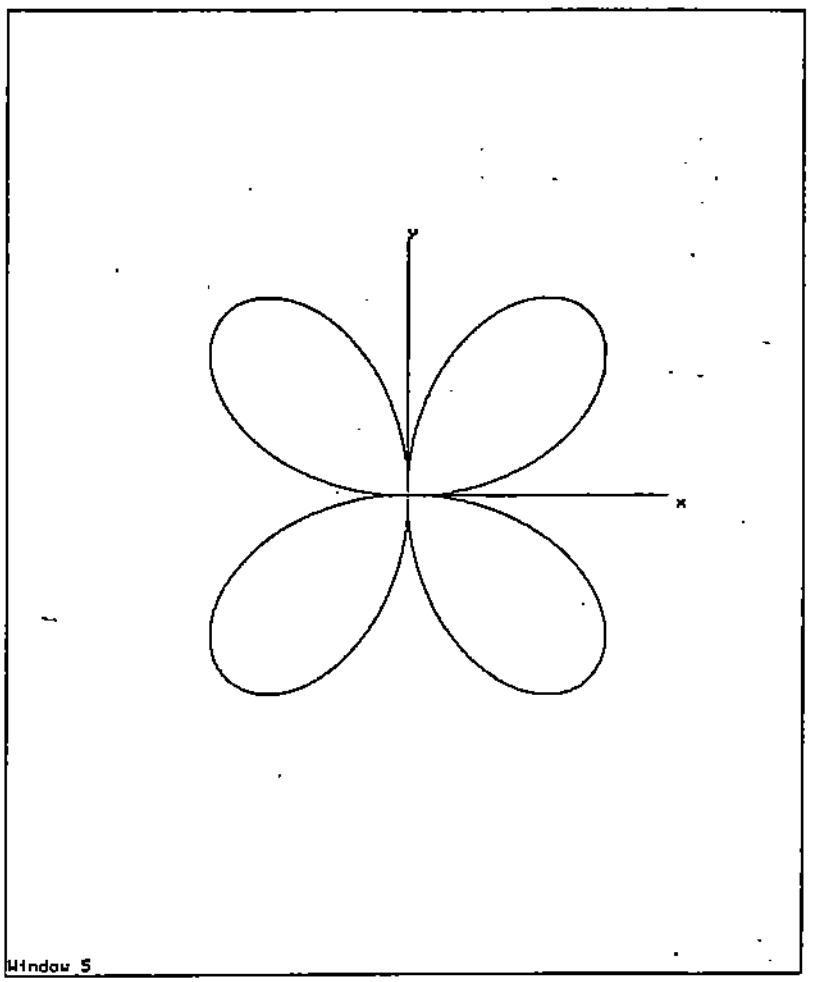

The distinguished polynomial of the given polynomial
1.0*Y^4 -1.0*Y^3 +2.0*X^2*Y^2 +3.0*X^2*Y +1.0*X^4
is
1.0*Y^3 -5.0*X^2*Y^2 -1.0*X^4*Y^2 -3.0*X^2*Y -1.0
*X^4*Y -1.0*X^4
RAT replaced -3.0 by -3/1 = -3.0
RAT replaced 1.0 by 1/1 = 1.0
RAT replaced -3.0 by -3/1 = -3.0
RAT replaced 1.0 by 1/1 = 1.0
A given polynomial -1.0*Y^3 +1.0*X^4 +2.0*X^2*Y^2 +
1.0*Y^4 +3.0*X^2*Y
= 0 has 3 factor(s) around the origin in real spac
e.
The following is a set of approximated factor(s) aro
und the origin :
1. Y = -0.33333325*T^2 -0.09641943*T^4
X = 1.0*T
2. Y = -1.7320508*T +2.6666667*T^2 -1.828276*
T^3 +0.54320973*T^4 +0.3661896*T^5
X = 1.0*T
3. Y = 1.7320508*T +2.6666567*T^2 +1.828276*
T^3 +0.54320973*T^4 -0.3661896*T^5
X = 1.0*T
$$$$ Would you like to continue?
(Yes or No) Yes
The corresponding (2,3) Pade approximants are
1. Y = 1.0*T^2
-3.0000007 +0.7777753*T^2
X = 1.0*T
2. Y = 1.0*T -0.56652397*T^2
-0.5773503 -0.56188614*T -0.25553102*T
^2 +0.018538008*T^3
X = 1.0*T
3. Y = 1.0*T +0.56652397*T^2
0.5773503 -0.56188614*T +0.25553102*T
^2 +0.018538008*T^3
X = 1.0*T
NIL
Dynamic Lisp Listener 3
Window 19
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Wed 4 Oct 3:52:55] bajaj CL USER: User Input
Y
X

Figure 2.2



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

Figure 3.1

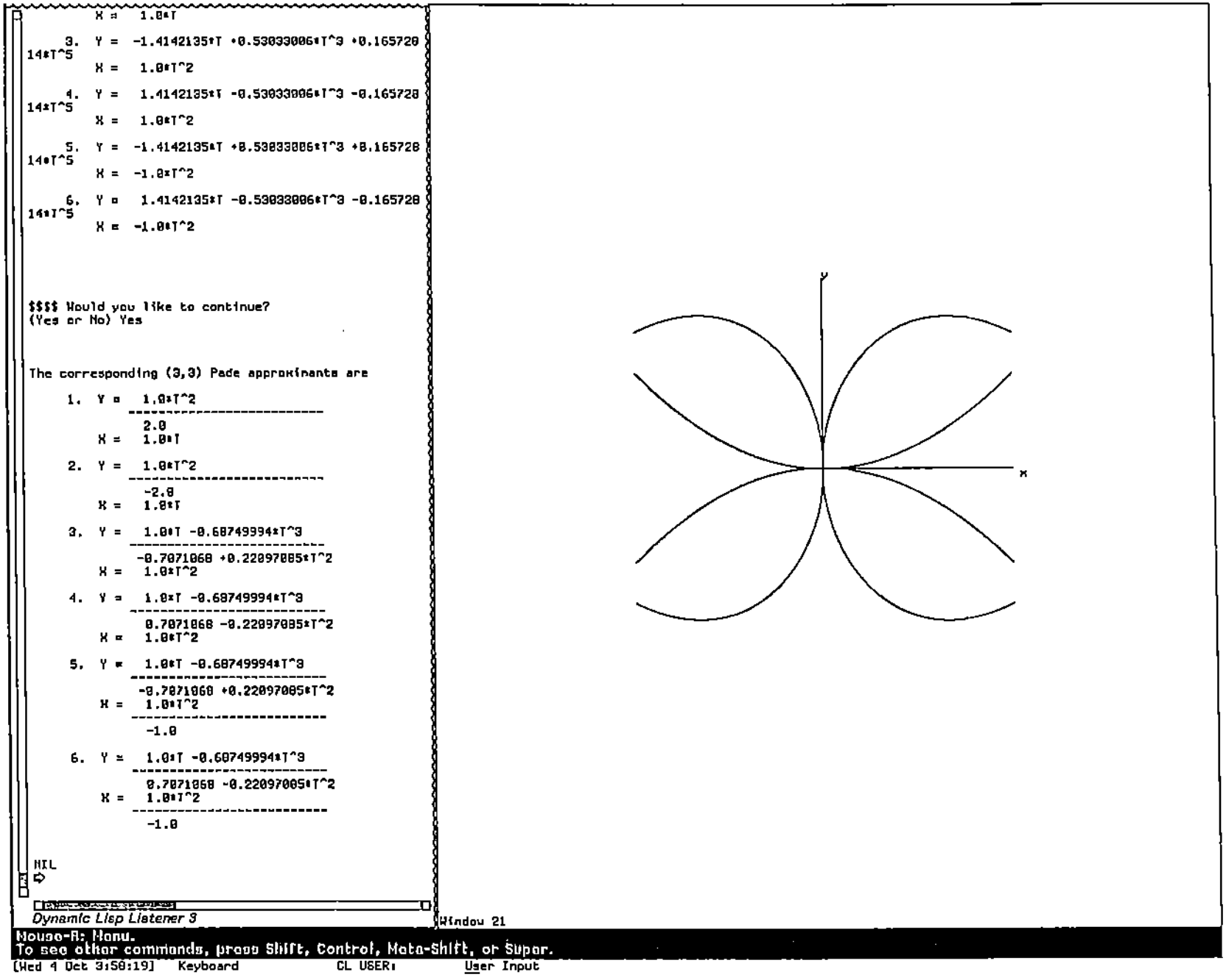

X = 1.0*T
3. Y = -1.4142135*T +0.53033006*T^3 +0.165720
14*T^5
X = 1.0*T^2
4. Y = 1.4142135*T -0.53033006*T^3 -0.165720
14*T^5
X = 1.0*T^2
5. Y = -1.4142135*T +0.53033006*T^3 +0.165720
14*T^5
X = -1.0*T^2
6. Y = 1.4142135*T -0.53033006*T^3 -0.165720
14*T^5
X = -1.0*T^2
$$$$ Would you like to continue?
(Yes or No) Yes
The corresponding (3,3) Pade approximants are
1. Y = 1.0*T^2
2.0
X = 1.0*T
2. Y = 1.0*T^2
-2.0
X = 1.0*T
3. Y = 1.0*T -0.68749994*T^3
-0.7071068 +0.22097085*T^2
X = 1.0*T^2
4. Y = 1.0*T -0.68749994*T^3
0.7071068 -0.22097085*T^2
X = 1.0*T^2
5. Y = 1.0*T -0.68749994*T^3
-0.7071068 +0.22097085*T^2
X = 1.0*T^2
-1.0
6. Y = 1.0*T -0.68749994*T^3
0.7071068 -0.22097085*T^2
X = 1.0*T^2
-1.0
NIL
Dynamic Lisp Listener 3
Y
X
Window 21
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Wed 4 Oct 3:50:19] Keyboard CL USER: User Input

Figure 3.2