

An Introduction to Parametrizing Rational Curves

Victoria Y.H. Wood

December 1, 2011

1. INTRODUCTION

To begin, the reader may wish to know what a rational curve *is*.

DEFINITION 1. A **rational curve** is an algebraic curve of genus 0. An alternative definition is that it is an algebraic curve C for which there exists a birational map $\mathbb{C} \rightarrow C$ given by $t \rightarrow (\chi_1(t), \chi_2(t)), \chi_1, \chi_2 \in \mathbb{C}(t)$; these two definitions are equivalent. Such a parametrization is called a rational parametrization of the curve; in this paper, "parametrization" will be understood to mean "[rational] parametrization".

Given that such a parametrization exists, the natural question might be "How might a parametrization of a given rational curve be constructed?" The book *Rational Algebraic Curves: A Computer Algebra Approach* ([1]) by F. Winkler and J. Sendra covers this topic in depth, in particular focusing on algorithms that may be used to construct a parametrization or, given a parametrization of a certain algebraic curve, determine certain useful properties about the parametrization. Most importantly, it describes the algorithm PARAMETRIZATION-BY-ADJOINTS, which is guaranteed to output a rational parametrization of a given rational algebraic curve in a finite number of steps.

One might ask for a motivation (ignoring any purely theoretical or aesthetic interests) for parametrizing rational curves. As it turns out, rational curves have applications in computer graphics, geometric modeling, solving Diophantine equations, and computing general solutions to first-order ordinary differential equations - to name just a few areas - so it is handy to know how to parametrize them.

As for myself, when I learned that curves of genus 0 were rationally parametrizable (and conversely, that any rationally parametrizable curve must have genus 0), I became interested in the subject and wished to know more. To my surprise, I found out a relatively simple algorithm existed to output a rational parametrization for any given curve of genus 0, and that it could even be worked through by hand for curves of low degree, as we will see in an example.

We now list a few useful definitions and lemmas, as given in [1]. (Proofs will be omitted.)

DEFINITION 2. An affine parametrization $P(t)$ of a rational curve C is **proper** if almost every point on C is generated by one value of the parameter t .

DEFINITION 3. A linear system of curves H **parametrizes** C iff

- 1) $\dim H=1$,
- 2) the intersection of a generic element of H and C contains a nonconstant point whose coordinates depend rationally on the free parameter in H , and
- 3) C is not a component of any curve in H .

DEFINITION 3A. An irreducible projective curve C is **parametrizable by lines** if there is a linear system of curves H of degree 1 (i.e. lines) that parametrize C .

LEMMA 1. Let $H(t)$ be a linear system of curves parametrizing C ; then, there is only one nonconstant intersection point of a generic element of $H(t)$ and C depending on t , and this gives a proper parametrization of C .

DEFINITION 4. The transformation Q of the projective plane $\mathbb{P}_2(\mathbb{K})$ given by $x' = yz, y' = xz$, and $z' = xy$ is called the **standard quadratic transformation**. A quadratic transformation is any transformation $Q \circ T$, where T is a change of coordinates. We note that, amongst other things, Q is its own inverse.

DEFINITION 5. The **first neighborhood** of a singular point P on C is the intersection of the image of C under the quadratic transformation taking P to $(0:0:1)$ and the line $z=0$, the second neighborhood of C is the union of the first neighborhoods of all the points in the first neighborhood of P , etc. The **neighborhood graph** of C is the union of all neighborhoods of C .

DEFINITION 6. A curve C' is adjoint to C iff $\text{mult}_P(Q_P(C')) \leq \text{mult}_P(Q_P(C)) - 1$ for every point P in the neighborhood graph of C , where $Q_P(C)$ is the sequence of quadratic transformations and linear transformations generating the neighborhood in the neighboring graph of C to which P belongs. In particular, for a curve C with only ordinary singular points, this condition is simple $\text{mult}_P(C') \leq \text{mult}_P(C)$ for all singular points P of C . $A_k(C)$ is the set of all adjoints of C of degree k .

LEMMA 2. Let C be a projective curve of degree d and genus 0, let $k \leq d-2, d-1$, and let S be a set of $kd - (d-1)(d-2) - 1$ simple points of C . Then $A_k(C) \cap H(k, \sum_{P \in S}(P))$ parametrizes C . If $k = d$ and S is a set of $d^2 - (d-1)(d-2) - 1 = 3(d-1)$ simple points of C , then choose $Q \notin C$; $A_d(C) \cap H(d, Q + \sum_{P \in S}(P))$ parametrizes C instead.

DEFINITION 7. If G is a polynomial in $K[x, y, z][t]$, we define $\mathbf{pp}_t(\mathbf{G})$ to be the primitive part of G with respect to t - that is, G divided by the g.c.d. of its coefficients.

DEFINITION 8. The length of an integer a is $\lfloor \log_2(|a|) \rfloor + 2$. $L_{MAX}(h), h \in [x_1, \dots, x_{r-1}][x_r]$ with coefficients $h_i \in \mathbb{Z}[x_1, \dots, x_{r-1}]$, is the length of $\|h\|_{max}$, where $\|h\|_{max} = \max_{i=0, \dots, \text{deg}(h)} \{\|h_i\|_{max}\}$. The length of a rational number $\text{frac}pq$, where $p, q \in \mathbb{Z}$ and $\text{GCD}(p, q) = 1$, is $\text{length}(p) + \text{length}(q)$.

DEFINITION 9. We say, for two functions $f(x)$ and $g(x)$ defined on a subset of the real numbers, $g(x) = \mathcal{O}(f(x))$ if, for x_0 sufficiently large, there exists sufficiently large $N > 0$ such that $|g(x)| < N|f(x)|$ for all $x > x_0$. The complexity of an algorithm is often described in terms of the time or number of steps it takes to carry out; if the amount of time it takes to carry out an algorithm is $\mathcal{O}(n^m + a_1 n^{m-1} + \dots + a_{m-1} n + a_m)$, $\mathcal{O}(\log_c(n))$, or $\mathcal{O}(c^n)$ (where n is part of the algorithm's input, $c > 0$, and $a_1, \dots, a_m \in \mathbb{C}$), we say that the complexity of the algorithm is polynomial, logarithmic, or exponential in n . (Alternatively, that the algorithm takes polynomial, logarithmic, or exponential time.)

Given these lemmas and definitions, we may now state the main algorithm in [1], which allows us to parametrize any given rational curve (though some are much simpler to parametrize than others).

2. PARAMETRIZATION BY ADJOINTS

Let an irreducible projective curve C of degree d and genus 0 be defined by the polynomial $F(x : y : z) \in \mathbb{K}[x, y, z]$.

1. If $d < 3$ or C has exactly one point of multiplicity $d - 1$, apply the following PARAMETRIZATION BY LINES algorithm:

If $d = 1$, C is a line. If $d > 1$, compute the $(d - 1)$ -fold point of C , and, perhaps after a change of variables, call it $(a : b : 1)$. Set $g(x, y) = F(x + a : y + b : 1)$. Let $g_d(x, y)$ and $g_{d-1}(x, y)$ be the homogenous components of $g(x, y)$ of degree d and degree $d - 1$, respectively. Return $P(t) = (-g_{d-1}(1, t) + ag_d(1, t), -tg_{d-1}(1, t) + bg_d(1, t), g_d(1, t))$.

2. Otherwise, choose $k \in \{d - 2, d - 1, d\}$ and compute the defining polynomial of $A_k(C)$.

3. Choose a set S of $kd - (d - 1)(d - 2) - 1$ simple points of C .

4. If $k < d$, compute the defining polynomial H of $A_k(C) \cap H(k, \sum_{P \in S} (P))$ - otherwise, choose $Q \notin C$ and compute the defining polynomial H of $A_d(C) \cap H(d, Q + \sum_{P \in S} (P))$.

5. Set one of the parameters in H to 1 and the other to t . Return the solution in $\mathbb{P}_2(\mathbb{K}(t))$ of $\{\text{pp}_t(\text{res}_y(F, H))=0, \text{pp}_t(\text{res}_x(F, H))=0\}$.

3. AN EXAMPLE OF THE APPLICATION OF PARAMETRIZATION BY LINES

Exercise 4.19 in [1]. Let C be the affine curve defined by $f(x,y)=(x^2+4y+y^2)^2-16(x^2+y^2)=0$. Compute a rational parametrization of C .

The homogenization of f is of course $g(x:y:z)=(x^2+4yz+y^2)^2-16z^2(x^2+y^2)$; evaluating this gives $(\frac{dg}{dx}, \frac{dg}{dy}, \frac{dg}{dz})=(2(2x)(x^2+4yz+y^2)-32z^2x, 2(4z+2y)(x^2+4yz+y^2)-32z^2y, 2(4y)(x^2+4yz+y^2)-32z(x^2+y^2))$. Setting $g, \frac{dg}{dx}, \frac{dg}{dy}$, and $\frac{dg}{dz}$ all equal to 0 gives that $2g-z\frac{dg}{dx}=2(x^2+y^2)(x^2+4yz+z^2)=0$, so either $x^2+y^2=0$ or $x^2+4yz+z^2=0$; letting $x^2+y^2=0$ immediately gives from $g=0$ that $(4yz)^2=0$, and letting $x^2+4yz+y^2=0$ gives from $\frac{dg}{dx}=\frac{dg}{dy}=0$ that $16z^2x=32z^2y=0$. $y=0$ gives $x=0$ immediately, and $z=0$ gives $x^2=(-1)y^2$, so the singular points are $(0:0:1)$, $(1:i:0)$, and $(i:1:0)$.

In this case, $k=d-2=2$ serves nicely, as then $\text{card}(S)$ simply is $(2)(4)-(3)(2)-1=1$, so we may choose any nonsingular point on $\{g=0\}$, say $(4:0:1)$.

All singularities are non-ordinary. Blowing up, we obtain that the points in the first neighborhood of $(0:0:1)$ are $(0:1:0)$ and $(1:0:0)$, both simple, that the points in the first neighborhood of $(1:i:0)$ are $(1:0:0)$ and $(1-4i:1:0)$, both simple, and that the points in the first neighborhood of $(i:1:0)$ are $(1:0:0)$ and $(1:4i:1:0)$, also both simple. As a result, we need not consider these neighboring points in the calculation of the adjoint curves.

Curves adjoint to $g=0$ then must at least pass through $\text{Sing}(\{g=0\})$, so we may write out $A_2(\{g=0\})$ as $\{H=a_{11}x^2+a_{22}y^2+a_{33}z^2+a_{12}xy+a_{13}xz+a_{23}yz \mid H(0:0:1)=H(1:i:0)=H(i:1:0)=0\}$. Then, $a_{33}=0$, $a_{11}-a_{22}+a_{12}i=0$, and $a_{22}-a_{11}+a_{12}i=0$, so $a_{12}=0$ and $a_{11}=a_{22}$. We then may write the general form of H as $a_{11}(x^2+y^2)+a_{13}xz+a_{23}yz$. Elements of $H(2,(4:0:1))$ pass through $(4:0:1)$, so $16a_{11}+4a_{13}=0$, and a generic element of $A_2(\{g=0\}) \cap H(2,(4:0:1))$ has the form $H=a_{11}(x^2+y^2)-4a_{11}xz+4a_{23}yz$.

Setting a_{11} equal to 1 and a_{23} equal to t , we obtain the form $H=x^2+y^2-4xz+tyz$. As $\{g=0\}$ is parametrized by $A_2(\{g=0\}) \cap H(2,(4:0:1))$, we may obtain the parametrization of g by solving $x^2+y^2-4xz+tyz=(x^2+4yz+y^2)^2-16z^2(x^2+y^2)=0$ for x, y , and z . This gives $x^2+y^2=4xz-tyz$, so $0=(4yz+4xz-tyz)^2-16z^2(x^2+y^2)=z^2((4x+(4-t)y)^2-(4x)^2-(4y)^2)=z^2((4x)^2+2(4x)(4-t)y+(4-t)^2y^2-(4x)^2-(4y)^2)=z^2(2(4x)(4-t)y+(4-t)^2y^2-(4y)^2)=z^2y(2(4x)(4-t)+(4-t)^2y-(16y))=z^2y(8(4-t)x+t(t-8)y)$. Hence, ignoring the powers of z and y , we see that we now need to solve $8(4-t)x+t(t-8)y=0$ and $z=\frac{x^2+y^2}{4x-ty}$. This gives $(x:y:z)=(y\frac{t(t-8)}{8t-32}:y:\frac{x^2+y^2}{4x+ty})=(y\frac{t(t-8)}{8t-32}:y:y\frac{(32-8t+t^2)^2}{-32t^2(t-4)})=(-4t^3(t-8):-32t^2(t-4):(16+(4-t)^2)^2)$. By substituting $4u+4=t$, we may simplify this to $(1024(-1+u^2)(1+u^2):-2048u(1+u^2):256(1+u^2)^2)$ and so, we have a rational parametrization of C :

$$\left(\frac{4(1+u)^2(u^2-1)}{(1+u^2)^2}, -\frac{8u^2(1+u^2)}{(1+u^2)^2}\right)$$

4. REMARKS ABOUT PARAMETRIZATION BY ADJOINTS

(As the computer algebraic system CASA was only available to those who had registered for a license, and the link to the license-registration was broken, I instead had to use Mathematica.)

For details on the computational complexity of the various steps of the algorithms, please see *On the Complexity of Parametrizing Rational Curves* by Mĭuk, Sendra, and Winkler ([3]), which discusses computational complexity of the various parts of the algorithm in depth. In particular, it shows the following result:

LEMMA 3. The worst-case complexity for the algorithm PARAMETRIZE (in essence, PARAMETRIZATION BY ADJOINTS), working on rational ordinary singularities, is $\mathcal{O}(d^{12}(d \log(d) + L_F)^2 + d^{12} \max\{L_R, \log(d) + L_S\}^2)$, where $L_F = L_{MAX}(F)$, L_R is the maximum length of the given rational simple points, and L_S is the maximum length of singularities.

Furthermore, [3] also proves that, if C is allowed to possess non-ordinary singularities, the time it takes to run the algorithm may be exponential in the degree of F ! This is an example of a problem for which it is much easier to deal with algebraic curves possessing only ordinary singular points.

Setting $k = d - 2$ obviously gives the quickest calculation, as fewer simple points of C need to be produced and the degrees of the polynomials that need to have their coefficients calculated are minimized.

Of these steps, computing defining polynomials is mechanically the simplest, as it is essentially solving a system of linear equations for the coefficients of the polynomials. The main difficulty lies in finding the points that members of $A_k(C) \cap H(k, \sum_{P \in S}(P))$ (or of $A_d(C) \cap H(d, Q + \sum_{P \in S}(P))$) must pass through, namely the singular points of C (and their neighboring points) and the $kd - (d - 1)(d - 2) - 1$ simple points lying on C . Fortunately, we need not search for each of these points individually.

We now introduce a definition which will be crucial to finding points in bulk:

DEFINITION 10. Let K be an algebraically closed field containing the coefficients of F , \mathbb{K} be the smallest subfield of K containing the coefficients of F , and let \mathbb{L} be a subfield of K such that $\mathbb{K} \subset \mathbb{L} \subset K$. The set of projective points $F = \{(p_1(\alpha) : p_2(\alpha) : p_3(\alpha)) \mid m(\alpha) = 0\} \subset \mathbb{P}^2(K)$ is called a **family of s conjugate points** over \mathbb{L} if the following conditions are satisfied:

- 1) $p_1, p_2, p_3, m \in \mathbb{L}[t]$, and $\gcd(p_1, p_2, p_3) = 1$,
- 2) m is square-free and $\text{degree}(m) = s$,
- 3) $\text{degree}(p_i) < \text{degree}(m)$ for $i = 1, 2, 3$,
- 4) F contains exactly s different points of $\mathbb{P}^2(K)$.

Such a family is denoted by $\{(p_1(t) : p_2(t) : p_3(t))_{m(t)}\}$.

Through exploitation of families of conjugate points, we need only determine $d - k + 1$ simple points on our own, as detailed in [1], by constructing families of conjugate simple points. In fact, as Bizzarri's paper ([2]) shows, only one simple point needs to be found on C in order for a family of $kd - (d - 1)(d - 2) - 1$ conjugate simple points on C to be constructed. (We will list the algorithm for this construction later.)

First, however:

Define C_a to be the affine algebraic curve corresponding to the image of the intersection of C with the hyperplane $\{z = 1\} \subset \mathbb{P}^2(K)$ under the isomorphism $z = 1 \rightarrow \mathbb{A}^2(K)$.

DEFINITION 11. We say an affine curve defined by a polynomial $f(x, y)$ is **in regular position with respect to x** if and only if the coefficient of $y^{\deg(f)}$ in f is nonzero and, if $f(x_0, y_0) = f(x_0, y_1) = \frac{df}{dx}(x_0, y_0) = \frac{df}{dx}(x_0, y_1) = 0, y_0 = y_1$.

We may check that C_a is in regular position with respect to x by checking that $\langle f(x, y_1), f(x, y_2), \frac{df}{dx}(x, y_1), \frac{df}{dx}(x, y + 2), 1 - w(y_1 - y_2) \rangle$ (where w is an added variable) has Gröbner basis equal to 1, so that, for $y_1 - y_2 \neq 0$, $f(x, y_0)$, $f(x, y_1)$, $\frac{df}{dx}(x, y_0)$, and $\frac{df}{dx}(x, y_1)$ are never all simultaneously 0.

If C_a is not in regular position with respect to x , we may perform a change of coordinates to make it be so. The usefulness of having C_a be in regular position with respect to x is that, in order to find $kd - (d - 1)(d - 2) - 1$ simple points and all the singular points on C_a , we need only find their x -coordinates, as then exactly one y -coordinate will correspond to each x -coordinate.

5. THE ALGORITHM IN MORE DETAIL

First, we compute the singular points of C_a .

Assuming C_a is in regular position with respect to x , we let $f(x,y)=F(x,y,1)$ and input the following code into Mathematica:

```
a[1]=(PolynomialGCD[Resultant[D[f, y], f, y], Resultant[D[f, x], f, y])
(*the roots of a[1] are exactly the x-coordinates of the singular points
of C_a *)
For[i=1; i < d, a[i + 1]]=(PolynomialGCD[c[i], Resultant[D[f, {y, i}],
f, y], Resultant[D[f, {y, i}], f, y]), i++)
(*the roots of a[i] are exactly the x-coordinates of the singular points of
C_a with multiplicity at least i+1*)
For[i1=1; i1 < d, b[i1]]=(a[i1]/PolynomialGCD[D[a[i1], x], a[i1]]),
i1++);b[d + 1]=1;
(*b[i] has exactly the same distinct roots as a[i], but with no multiple roots*)
For[i2=1; i2 < d, c[i2]]=(b[i2]/b[i2 + 1]), i2++)
(*the roots of c[i] are exactly the x-coordinates of the singular points of C_a
with multiplicity at least i+1 but less than i+2, i.e. those of multiplicity
exactly equal to i+1*)
For[i3=1; i3 < d, gb[i3]]=(GroebnerBasis[f, D[f, x], D[f, y],
c[i3]], {x, y}), i3++)
(*this step gives the Gröbner basis gb[i3] of the ideal corresponding
to the set of singular points of C_a with multiplicity i3+1*)
```

Since we have ensured there exists exactly one simple intersection point for every singular point of f , the computed Gröbner basis $gb[j]$ contains a polynomial S that is linear in y , as there is exactly one common intersection point for each root of f . Write this polynomial as $r_j(x, y) = u_j(x) + v_j(x)y$; we now obtain the set of j -fold singular points of C_a as a family of conjugate points $F_j = \{(\alpha : \frac{u_j(\alpha)}{v_j(\alpha)} : 1) \mid c_j(\alpha) = 0\}$. Finding the rest of the singular points of C (those with z -coordinate 0) consists of checking the zeroes of $F(x : y : 0)$ to see if they are singular points of C , and if so, of what multiplicity. Denote the set of j -fold singular points of C with z -coordinate 0 as F'_j .

Now that we have obtained the singular points, we may find the adjoint curves of degree d to C . For curves with only ordinary singular points, we solve the system of linear equations

$$\begin{aligned} \sum_{0 < j_1, j_2 < k} r_{j_1, j_2} x^{j_1} y^{j_2} &= 0 \text{ for all } (x, y) \in \bigcup_{1 < j < d} F_j; \\ \frac{d}{dx} \left(\sum_{0 < j_1, j_2 < k} r_{j_1, j_2} x^{j_1} y^{j_2} \right) &= 0 \text{ for all } (x, y) \in \bigcup_{2 < j < d} F_j; \\ \dots; \\ \frac{d^i}{dx^i} \left(\sum_{0 < j_1, j_2 < k} r_{j_1, j_2} x^{j_1} y^{j_2} \right) &= \frac{d^i}{dx^{i-1}y} \left(\sum_{0 < j_1, j_2 < k} r_{j_1, j_2} x^{j_1} y^{j_2} \right) = \dots \\ &= \frac{d^i}{dy^i} \left(\sum_{0 < j_1, j_2 < k} r_{j_1, j_2} x^{j_1} y^{j_2} \right) = 0 \text{ for all } (x, y) \in \bigcup_{i+1 < j < d} F_j; \\ \dots; \\ \sum_{0 < j_1 < d} r_{j_1, d-j_1} x^{j_1} y^{d-j_1} &= 0 \text{ for all } (x : y : 0) \in F'_j \dots \\ \frac{d^i}{dx^i} \left(\sum_{0 < j_1 < d} r_{j_1, d-j_1} x^{j_1} y^{d-j_1} \right) &= \frac{d^i}{dx^{i-1}y} \left(\sum_{0 < j_1 < d} r_{j_1, d-j_1} x^{j_1} y^{d-j_1} \right) = \dots \\ &= \frac{d^i}{dy^i} \left(\sum_{0 < j_1 < d} r_{j_1, d-j_1} x^{j_1} y^{d-j_1} \right) = 0 \text{ for all } (x, y) \in \bigcup_{i+1 < j < d} F_j; \dots \end{aligned}$$

for the linear relations between the r_{j_1, j_2} to obtain the generic form of an adjoint curve of C_a . Once we have done that, we may proceed to find the $kd - (d-1)(d-2) - 1$ simple points.

Take a simple point $P=(p_1 : p_2 : p_3)$ on $C=F(x : y : z)=0$. Let us take a degree- k adjoint curve to C that also passes through P , and call this curve M ; let

M be defined by some homogenous polynomial $H(x : y : z)$. Let $f(x, y) = F(x : y : 1)$ and $h(x, y) = H(x : y : 1)$. By Bezout's Theorem and the rationality of C , the intersection of C_a and M_a (given by $\{f(x, y) = 0\}$ and $\{h(x, y) = 0\}$, respectively) should contain $kd - (d-1)(d-2)$ simple points if $p_3 \neq 0$ and $kd - (d-1)(d-2) - 1$ simple points if $p_3 = 0$ (as $(\frac{p_1}{p_3}, \frac{p_2}{p_3})$ does not lie in $C_a \cup M_a$). We now may input the following code into Mathematica:

```

h:=(PolynomialMod[H, {z-1}])
A1:=(Resultant[f, h, y])
A2:=(PolynomialGCD[Resultant[f, D[f, x], y], Resultant[f, D[f, y], y]])
B1:=(A1/PolynomialGCD[A1, D[A1, x]])
B2:=(A2/PolynomialGCD[A1, D[A2, x]])
(*A1 has roots exactly equal to the x-coordinates of the elements of  $C_a \cup M_a$ ,
while A2 has roots exactly equal to the x-coordinates of the singular
points of  $C_a$ . B1 and B2 have the same properties, but are square-free.*)
E:=(B1/B2)
If  $p_3 \neq 0$  ( $g = e / (x - p_1 / p_3)$ )
(*the above is to remove  $(\frac{p_1}{p_3}, \frac{p_2}{p_3})$  from the list of simple points*)
Else  $g = e$ 
(*the roots of  $g$  are the x-coordinates of the elements of  $C_a \cup M_a$  that are
not singular points of  $C_a$ , namely the simple intersection points of  $C_a$  with
 $M_a$ *)

```

The g constructed has roots exactly corresponding to the x -coordinates of the simple intersection points of C_a with M_a . It is possible that some of the intersection points of C and M in projective space will lie on the line at infinity or have the same x -coordinate when mapped into affine space; in this case, g will not have full degree $kd - (d-1)(d-2) - 1$, and a new M should be selected. Otherwise, compute GroebnerBasis[f, h, g]. Since we have ensured there exists exactly one simple intersection point for every root of g , the computed basis contains a polynomial p that is linear in y , as there is exactly one common intersection point for each root of g . Write this polynomial $p(x, y)$ as $q(x) + s(x)y$.

Then, $F = \{(\beta : q(\beta) / s(\beta) : 1)_{g(\beta) = 0}\}$ is a family of $kd - (d-1)(d-2) - 1$ simple points on C_a , as desired.

Then, we add the condition
 $\sum_{0 < j_1, j_2 < k} r_{j_1, j_2} x^{j_1} y^{j_2} z^{k - (j_1 + j_2)} = 0$ for all $(x : y : 1) \in F$.

and solve for the further linear relations between the r_{j_1, j_2} . We will be left with an equation that is linear in t , and we then go about the process of solving for the generic point in $A_k(C) \cap H(k, \sum_{P \in S} (P))$ (or $A_d(C) \cap H(d, Q + \sum_{P \in S} (P))$). We may do this by resultants, as the algorithm suggests, or, in small cases, solve the resulting equations by hand, as in the example.

6. SUMMARY

We have showed how to obtain a rational parametrization for any given curve of genus 0. However, this algorithm may be quite slow for large n , as [3] shows. Thanks to usage of families of conjugate points, the greatest slowing point in the algorithm, when calculations are aided by computer, is not in finding the simple points or even in finding the singular points, but finding the neighborhood graph of C . On a CAS better-adapted to dealing specifically with algebraic curves, such as SINGULAR or CASA, this is likely a simple command away (even though it would still take a while to compute). However, as finding the neighborhood graph would be a complicated procedure, due to having to take the correct quadratic transformation for every non-ordinary singular point, finding the points of intersection, checking to see if any of the resulting points in the first neighborhoods were non-ordinary, etc., I elected to omit the algorithm from this paper. It may be found in [3] under the name NEIGHBORHOOD TREE.

I split up the algorithms as I was not sure how to induce Mathematica to recognize and select the elements of the Gröbner bases that were linear in y . If I had known a way to do this, I would have joined the two blocks of code together by computing F_j in the code itself.

If I were to continue learning about parametrization of rational curves, I would like to learn more about interesting variations on rational curves, such as what can be done with rational curves possessing only ordinary singular points or almost-rational curves. I would also like to know what work has been done on rationally parametrizable surfaces; there is a great deal in this field to explore, and I would enjoy learning more about it.

7. BIBLIOGRAPHY

- [1] Sendra, J., Winkler, F., Perez-Diaz, S.: *Rational Algebraic Curves: A Computer Algebra Approach*. Springer-Verlag, New York, 2007.
- [2] Bizzarri, M., Lávička, M.: *Algorithm for parametrization of rational curves revisited*. Journal for Geometry and Graphics, 2011.
- [3] Mňuk, M., Sendra, J., Winkler, F.: *On the Complexity of Parametrizing Curves*. Contributions to Algebra and Geometry, vol. 37 (1996), No. 2, pp. 309-328.

8. FIGURES

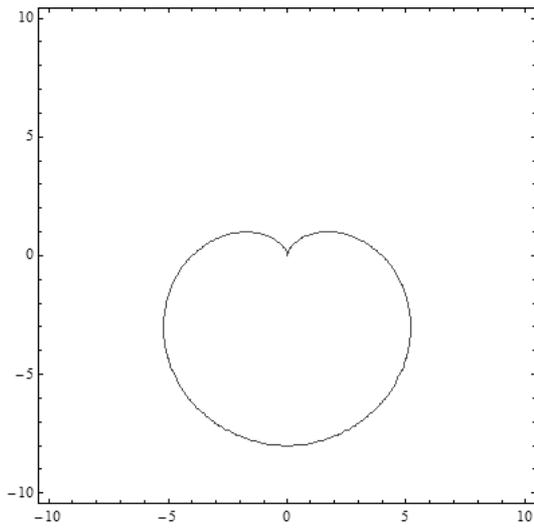


Figure 1: The contour plot of the curve in the example for $-10 < x, y < 10$

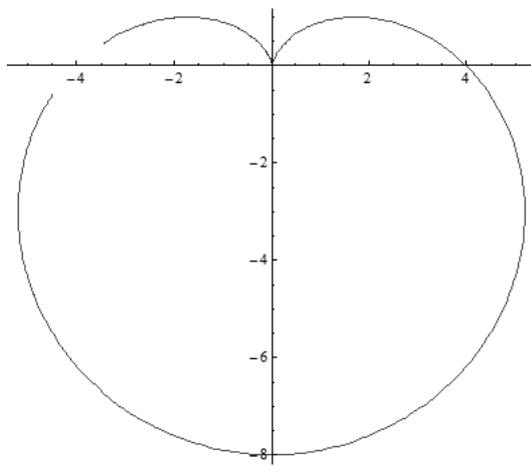


Figure 2: The plot of the parametrization produced in the example for $-15 < u < 15$