Computing parametrizations of rational algebraic curves

Mark van Hoeij
Department of mathematics
University of Nijmegen
6525 ED Nijmegen
The Netherlands
e-mail: hoeij@sci.kun.nl

June 1, 1994

Abstract

In this paper I want to present a new method for computing parametrizations of algebraic curves. Basically this method is a direct application of integral basis computation. Examples show that this method is faster than older methods.

1 Introduction

Suppose we are given two rational functions $X(t)$ and $Y(t)$, e.g. $X(t) = 2t/(t^2+1)$ and $Y(t) = (t^2-1)/(t^2+1)$. Then the set of points $[X(t), Y(t)]$ where t runs through the complex numbers defines an algebraic curve in the plane, in this particular example a circle. We call $[X(t), Y(t)]$ a parametrization of this curve.

We can also represent an algebraic curve by a polynomial equation $f(x,y) = 0$, e.g. $x^2 + y^2 - 1 = 0$. Algebraic curves given by a parametrization can also be represented by a polynomial, but not necessarily vice versa. The theory of algebraic curves tells us that a curve can be parametrized precisely when the curve is rational, i.e. when the genus of the curve is zero. However, in most textbooks it is not mentioned how a parametrization can be found.

Different methods for computing parametrizations are known. The method described in [5] is implemented in the Maple package CASA. Also a good description of the problem and applications are given in [5].

To find a parametrization we first compute a parameter. A parameter is an element of the function field that has precisely one pole on the curve, with multiplicity one.

The problem of finding parametrizations in minimal algebraic extensions is not treated in this paper. This problem is reduced to the problem of finding one point on the curve in a minimal algebraic extension, which is treated in [3] and [6]. A point can be found using an algebraic extension on the coefficients field of degree two. At this moment my implementation does not yet make use of these methods. The current version only tries to find a good point on lines through singularities. In the worst case a point will be used in an algebraic extension of degree equal to the degree of the curve.

2 Brief outline

In this section we will sketch the algorithm. We leave the details for the next section. We use the following notations:

- L is an algebraically closed field of characteristic 0.
- $f(x,y) \in L[x,y]$ is irreducible.
- n is the degree of f in y.
- C is the projective algebraic curve given by f.
- $L(C)$ is the function field of the curve. We can identify this field with $L(x)[y]/(f)$. We will denote elements of this field as polynomials in y of degree $< n$.
- F is the homogenization of f. It is an element of $L[x,y,z]$ of minimal degree such that $f = F_{z=1}$ (F with $z = 1$ substituted). We can identify the function field $L(C) = L(x)[y]/(f)$ with $L(z)[y]/(F_{x=1})$. Writing an element a of $L(x)[y]/(f)$ as element of $L(z)[y]/(F_{x=1})$ is done by making a homogeneous followed by a substitution $x = 1$.
- p is a parameter. That means p generates the function field, $L(p) = L(C) = L(x)[y]/(f)$.
- a parametrization is a list $[X(t), Y(t)]$ of rational functions such that $f(X(t), Y(t)) = 0$ and such that $L(X(t), Y(t)) = L(t)$.

An rational algebraic curve allows a parametrization. To find this parametrization we first compute a parameter p. Then we can express x and y in $L(C) = L(p)$ as rational functions in p. This gives us a parametrization.

A parameter is characterized by the property that it has only one pole, of multiplicity 1, in the places of the curve C. To find this parameter we divide the projective plane into two disjoint parts, A and B. Then we compute a starting function P, that has only 1 pole of multiplicity 1 in $A \bigcap C$. So P has the desired property in the part $A \bigcap C$, but it may still have poles in $B \bigcap C$.

Now we want to adapt P in such a way that the one pole in $A \bigcap C$ remains, and that the poles in $B \bigcap C$ disappear. We do this by adding a function Q. We compute a Q satisfying following conditions:

- Q has no poles in $A \bigcap C$
- $P + Q$ has no poles in $B \bigcap C$.

If Q satisfies these conditions we see that $p = P + Q$ has precisely 1 pole on the curve. Then we know that p is a parameter. We use here that f is irreducible. Also the genus must be zero, otherwise the curve is not rational and no parameter exists. We can let the algorithm check these two requirements.

Now we want to express x and y as rational functions in p. Since p is algebraic over $L(x)$ we can compute a polynomial relation between x and p. In principle this can be done as follows. Suppose $p = p_1/p_2$ where p_1 and p_2 are elements of $L[x,y]$. Then we can compute the resultant $\mathrm{Res}_y(tp_2 - p_1, f)$. This gives a polynomial in x and t which vanishes for $t = p$. We can solve x from this relation and express it as a rational function in t. This is possible because $x \in L(p)$.

In practise, however, this resultant computation can be a bottleneck for the algorithm. A faster computation goes as follows. Substitute three different generic numbers for x and compute the resultant for all three values. Use the fact that the degree of the numerator and the denominator of the rational function $X(t)$ are bounded by n. $X(t)$ can be reconstructed from these three resultant computations. $Y(t)$ can be computed the same way. For efficiency reasons we always use small integers as generic numbers.

3 The algorithm

If the curve C contains the point $(0, 1, 0)$ we will remove this point by substituting $x + iy$ for x in f where i is a generic integer. After that we may assume that $(0, 1, 0)$ is not a point on the curve. Then we can divide the projective plane, except the point $(0, 1, 0)$, in the x-y plane and the line $z = 0$ in the z-y plane (the line at infinity). For the splitting of the plane in parts A and B there are different possibilities. We choose A is the x-y plane and B is the line $z = 0$ for the rest of this paper. Define $O_A \subset L(x)[y]/(f)$ as the ring of all functions having no poles on A, and O_B as the ring of all functions having no poles on B. Elements of O_B will be denoted as elements of $L(z)[y]/(F_{x=1})$.

Then a starting function P must be a function having precisely one pole on C in the x-y plane. Given a regular point on the curve it is not difficult to compute such a function P. However, if we need algebraic extensions of the coefficients field to find this point, these algebraic extensions will appear in the resulting parametrization. This makes the result ugly and the parametrization slow, especially the computation in section 3.2. Finding a good point is therefore an important problem. As was said before this problem is not yet properly handled by my implementation. It lets a user specify a point. If no point is specified, like in the test examples in section 4, it looks for good points on lines through singularities. If no good point was found then the algorithm will pick a point on the line $x = i$ where i is an integer. This will generally require an algebraic extension of degree n.

3.1 Finding the function Q

For a description of integral bases and computation of integral bases I refer to [4]. The method used here to compute integral bases is by using Puiseux expansions. The speed of this method depends on the bound given in [4] for the Puiseux expansions and the bound for the maximal denominator in the integral basis.

There are two requirements on Q. The first is that Q has no poles on C in the x-y plane, i.e. $Q \in O_A$. Because $(0,1,0)$ is not a point on the curve f we have that y is integral over $L[x]$ (or more precise: the class of y in L(C)=$L(x)[y]/(f)$ is integral over $L[x] \subset L(C)$). The fact that Q has no poles in the x-y plane is equivalent with Q being integral over $L[x]$. When we have computed an integral basis for O_A, we know in theory all such functions. We will show by an example how one can check if a given function is integral over $L[x]$.

Suppose $f = y^4 - 2xy^2 + (2x^3 - 6x + 4)y + x^4 - 2x^2 - 4x + 2x^3 + 4$ and we want to check if a given function $v \in L(C)$ is integral. For complicated v the fastest method of checking whether a given function is integral is by computing an integral basis. The integral basis for this f is

$$b = [1, y, \frac{y^2-1}{x-1}, \frac{-2x+3-2xy+y-y^2+y^3}{x^2-1}].$$

That means that

$$O_A = L[x]b = L[x]1 + L[x]y + L[x]\frac{y^2-1}{x-1} + L[x]\frac{-2x+3-2xy+y-y^2+y^3}{x^2-1}.$$

Checking if v is an element of $L[x]b$ is done the same way as e.g. one checks if $x^2+x \in L[x](x+1)$, namely by computing the remainder modulo the elements in the basis. We will show this by example.

For a given function $v = (y^3 + a_1 y + a_2)/(x-1)$ having indeterminates as coefficients, we can compute linear relations in the indeterminates a_1 and a_2 equivalent with this function being integral as follows. First remove the denominators.

$$v \in L[x]b \iff v \equiv 0 \mod b \iff (x^2-1)v \equiv 0 \mod (x^2-1)b$$

Multiplying out denominators $[b_1, b_2, b_3, b_4] = (x^2-1)b$ yields

$$[x^2-1, (x^2-1)y, (x+1)(y^2-1), -2x+3-2xy+y-y^2+y^3]$$

and

$$V \overset{\text{def}}{=} (x^2-1)v = (x+1)(y^3 + a_1 y + a_2).$$

We start the computation of the remainder modulo $[b_1, b_2, b_3, b_4]$ with the highest degree in y, which corresponds to the last element in the integral basis. Note that the degrees in y in the integral basis are always $0, 1, \ldots, n-1$.

$$\begin{aligned}
V - (x+1)b_4 &= r_4 = & (x+1)(2x+2xy-3+a_1y+a_2-y+y^2) \\
r_4 - b_3 &= r_3 = & (x+1)(2x+2xy-y+a_1y+a_2-2) \\
r_3 - 2b_2 &= r_2 = & a_1xy + a_2x + a_1y + a_2 + 2x^2 + xy - 2 + y \\
r_2 - 2b_1 &= r_1 = & (a_1+1)xy + (a_1+1)y + a_2x + a_2
\end{aligned}$$

So the remainder of V modulo $[b_1, b_2, b_3, b_4]$ is r_1. The coefficients of r_1 are a_1+1, a_1+1, a_2 and a_2. So we can find all values of a_1 and a_2 for which

v is integral by solving the following dependent system of linear equations: $a_1 + 1 = 0$, $a_1 + 1 = 0$, $a_2 = 0$ and $a_2 = 0$.

This shows how the condition "Q has no poles in part A" can be translated into linear equations. In almost the same way we can translate "$P + Q$ has no poles in part B", or equivalently $P + Q \in O_B$, into linear equations. The difference is that this time we need to have an integral basis for O_B. $F_{x=1}$ describes the curve in the z-y plane. In this plane we consider the line $z = 0$. Now y (more precise: the class of y in $L(z)[y]/(F_{x=1})$) is integral over the local ring $L[z]_{(z)}$ ($L[z]$ localized in the ideal (z)) because $(0, 1, 0)$ is not a point on the curve. We compute a local integral basis $[c_1, \ldots, c_n]$, i.e. a basis for all elements which are integral over the ring $L[z]_{(z)}$. So $O_B = L[z]_{(z)}c_1 + \ldots + L[z]_{(z)}c_n$.

Using this basis $[c_1, \ldots, c_n]$ we can check if a function v has poles on the line $z = 0$. The only difference with the above example is, that now we work with a local ring $L[z]_{(z)}$ instead of $L[x]$. In the algorithm the only difference is that first all factors in $L[z]$ in the denominator of v except powers of z are multiplied out.

We will write an ansatz for Q with indeterminates a_{ij} as coefficients.

$$Q = \frac{\sum_{j=0}^{n-1} \sum_{i=0}^{N-j} a_{ij} x^i y^j}{k}$$

where N is an integer and $k \in L[x]$. We need a bound for N and a bound for the denominator k. A bound for the denominator is a polynomial $K \in L[x]$ such that $k|K$. Such a bound K is given by the largest denominator in the integral basis for the x-y plane.

The number N – degree(k) is equal to the multiplicity of z in the denominator, if we make Q homogeneous and substitute $x = 1$. The maximal denominator of the local integral basis $[c_1, \ldots, c_n]$ in the the z-y plane is a bound for the multiplicity of z in the denominator of $P + Q$, because $P + Q \in L[z]_{(z)}c_1 + \ldots + L[z]_{(z)}c_n$. This implies a bound for the multiplicity of z in the denominator of Q (homogeneous and $x = 1$ substituted). This gives a bound for N.

Summary of the computation of p:

1. If $(0, 1, 0)$ is a point on the curve, i.e. if the degree of the curve is larger than the degree of f in y, apply a linear transformation to remove this point from the curve.

2. Compute an integral basis $[b_1, \ldots, b_n]$ for O_A and a local integral basis $[c_1, \ldots, c_n]$ for O_B. So $O_A = L[x]b_1 + \ldots + L[x]b_n$ and $O_B = L[z]_{(z)}c_1 + \ldots + L[z]_{(z)}c_n$.

3. Write $Q = (\sum_{j=0}^{n-1} \sum_{i=0}^{N-j} a_{ij} x^i y^j)/K$. Here K is the maximal denominator in $[b_1, \ldots, b_n]$. Computation of N: Let z^d be the maximal denominator in $[c_1, \ldots, c_n]$. Then the multiplicity of z in the denominator of $P + Q$ is at most d. So the multiplicity of z in the denominator of Q is at most the maximum of d and the multiplicity of z in the denominator of P. Now take N as the sum of degree(K) and this maximum.

4. Compute linear equations for $Q \in O_A$ and $P+Q \in O_B$ and find a solution of these equations. Substitute this solution for a_{ij} in Q.

5. return $P+Q$

3.2 Expressing x and y as rational functions in p

In the previous section we have seen that a parameter p can be found using integral basis computation and solving linear equations. Note that apparently we were able to find a parameter without investigating the curve and the singularities. This is not true though, the integral basis algorithm does compute singularities and Puiseux expansions. But we do see here that an integral basis, plus a local integral basis at infinity, gives very useful information about the curve. It gives sufficient information to find a parameter, given a starting function P. What we still need to do now is express x and y as rational functions in this parameter. We will only show this for y.

Let $p = p_1/p_2$ where p_1 and p_2 are elements of $L[x,y]$ (in fact we have $p_2 \in L[x]$ in our algorithm, but we do not use that here). Call n_x the degree of f in x. Now $Y(t)$ can be written as a/b where a and b are polynomials in t of degree $\leq n_x$. A generic line $y = i$ intersects the curve in n_x different points in the x-y plane. In all these points the parameter p takes different values, call them v_j, $j = 1 \ldots n_x$. We denote f with $y = i$ substituted by $f_{y=i}$. We can compute $w = \prod(t - v_j)$ because w equals a constant times $R_i = \mathrm{Res}_x(t(p_2)_{y=i} - (p_1)_{y=i}, f_{y=i})$. This latter statement follows because w and R_i both have degree $\leq n_x$ and have the same roots v_j, $j = 1 \ldots n_x$, which are all different for a generic i. Because $Y(v_j) = i$, $j = 1 \ldots n_x$ the numerator of $Y(t) - i$ must have v_j as zeros, hence $a - ib$ is equal to w times a constant. If we have this information for three different values i then we can compute a/b.

So we determine a/b using the fact that we know $a - ib$ up to a constant for three different values of i. Because this system is overdetermined (except for $n_x = 1$) we can also use this as an automatic check for the correctness of the result. From two resultant computations R_{i_1} and R_{i_2} for generic integers i_1 and i_2 we can conclude that $a/b = (-i_1 c R_{i_2} + i_2 R_{i_1})/(R_{i_1} - c R_{i_2})$ for some unknown constant c. For this, these i_j do not need to be generic in the sense that R_{i_1} and R_{i_2} are squarefree. Having degree$(R_{i_j}) = n_x$ is sufficient.

Summary of the computation of $Y(t) = a/b$ from p:

1. Take three different integers i_j such that $R_{i_j} = \mathrm{Res}_x(t(p_2)_{y=i_j} - (p_1)_{y=i_j}, f_{y=i_j})$ has degree n_x for $j = 1 \ldots 3$. Then $R_{i_j} = c_j(a - i_j b)$ for some unknown constants c_j.

2. $(-i_1 c R_{i_2} + i_2 R_{i_1}) - i_3(R_{i_1} - c R_{i_2})$ must be equal to R_{i_3} up to a constant factor. Solve c from this and return $(-i_1 c R_{i_2} + i_2 R_{i_1})/(R_{i_1} - c R_{i_2})$

4 A few tests

I have compared my implementation (IntBasis) with the Maple package CASA, version 2.1. This package uses a method described in [5]. The syntax for

IntBasis is:

```
        genus(f,x,y,‘parametrization‘,t);
```

and for CASA the syntax is:

```
        a:=mkImplAlgSet([f],[x,y]);
        impl2para(a,t);
```

I have tried four tests in Maple V release 2 on a workstation, without specifying a point on the curve. If a good point were specified the computation would probably be faster. The first two examples are copied from [5]. Both implementations use no algebraic numbers to denote the a parametrization for these two. The third example is of degree 5 in y and of degree 2 in x. Both implementations use quadratic algebraic numbers for this example. The last one is of degree 8 in both x and y. My implementation uses algebraic numbers of degree 4 for f_4, though we know that it should be possible using an extension of at most degree 2.

$$f_1 = y+x+2\,y^3+\frac{13\,xy^2}{4}+\frac{13\,x^2y}{3}+6\,x^3+y^5+\frac{9\,xy^4}{4}+4\,x^2y^3+9\,x^3y^2+4\,x^4y+9\,x^5$$

$$f_2 = \tfrac{1251\,y^4}{115}+\tfrac{5184\,xy^3}{115}+\tfrac{5354\,x^2y^2}{115}+x^4-\tfrac{9552\,xy^4}{115}-\tfrac{22496\,x^2y^3}{115}-\tfrac{5424\,x^3y^2}{115}-\tfrac{32\,x^4y}{23}$$
$$+192\,x^2y^4+\tfrac{17472\,x^3y^3}{115}-\tfrac{13824\,x^3y^4}{115}$$

$$f_3 = 1805\,y^5+(3610\,x+3610)\,y^4+(-2703\,x^2-12626\,x-13533)\,y^3$$
$$+(5406\,x^2+16218\,x+10812)\,y^2+(-4508\,x^2-18032\,x-16227)\,y$$
$$+3610\,x^2+14440\,x+14440$$

$$f_4 = y^8+(8\,x+14)\,y^7+(28\,x^2+102\,x+84)\,y^6$$
$$+(56\,x^3+318\,x^2+546\,x+282)\,y^5$$
$$+(70\,x^4+550\,x^3+1476\,x^2+1590\,x+576)\,y^4$$
$$+(56\,x^5+570\,x^4+2124\,x^3+3582\,x^2+2706\,x+720)\,y^3$$
$$+(28\,x^6+354\,x^5+1716\,x^4+4030\,x^3+4770\,x^2+2646\,x+518)\,y^2$$
$$+(8\,x^7+122\,x^6+738\,x^5+2264\,x^4+3740\,x^3+3252\,x^2+1326\,x+184)\,y$$
$$+x^8+18\,x^7+132\,x^6+508\,x^5+1101\,x^4+1338\,x^3+854\,x^2+244\,x+25$$

Computation times in seconds: (unknown means no answer after 3 days)

	IntBasis	CASA 2.1
f_1	10	32
f_2	17	2147
f_3	26	3313
f_4	311	unknown

In the last example the parameter computation takes 41 seconds. The rest of the time is mainly spent on resultant computations needed to express x and y as rational functions of the parameter. A good implementation of a resultant algorithm that can work with algebraic numbers would greatly speed up the computation in this example.

Availability: CASA 2.1 is available at ftp.risc.uni-linz.ac.at. IntBasis can be found in the Maple share library at neptune.inf.ethz.ch. The next share library will contain version January 1994 of IntBasis. This version contains the parametrization code. IntBasis and my paper [4] are also available by e-mail request (hoeij@sci.kun.nl).

Final remarks (14 april 1994): At the moment a new method is being implemented in CASA which is faster than CASA 2.1. I was sent computation times and also new test examples by a referee, showing that the running times of CASA's new implementation are similar to those of IntBasis. Furthermore CASA attemps to find parametrizations in minimal algebraic extensions. In one of the new examples IntBasis was 5 times slower, in an other example 27 times faster. These latter tests were run on different platforms though, because I do not have the new code.

References

[1] S.S. Abhyankar, C.L. Bajaj, *Automatic parametrization of rational curves and surfaces III: Algebraic plane curves*, Computer Aided Geometric Design 5, 309-321 (1988)

[2] G.A. Bliss, *Algebraic Functions*, Dover (1966)

[3] D. Hilbert, A. Hurwitz, *Ueber die Diopantischen Gleichungen vom Geschlecht Null*, Acta math 14, 217-224 (1890)

[4] Mark van Hoeij, *An algorithm for computing an integral basis in an algebraic function field*, Submitted to J. Symbolic Computation.

[5] J.R. Sendra, F. Winkler, *Symbolic parametrizations of curves*, J. Symbolic Computation 12, No. 6, 607-631 (1991)

[6] J.R. Sendra, F. Winkler, *Determining Simple Points on Rational Algebraic Curves*, Technical Report RISC 93-23 University Linz (1993)

[7] R.J. Walker, *Algebraic curves*, Princeton University Press. (1950)