An algorithm for computing the Weierstrass normal form

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

April 9, 1995

Abstract

This paper describes an algorithm for computing a normal form $y^2 + x^3 + ax + b$ for algebraic curves with genus 1. The corresponding isomorphism of function fields is also computed.

1 Introduction

The goal of the present paper is to compute a normal form for algebraic functions and curves. In [11] and [9] the case $g = 0$ is treated. This paper treats the case $g = 1$. These normal forms can be applied to the problem of integrating algebraic functions. Integration in general is a very hard problem. Because of this it is useful to have alternative methods for special cases. In [2, 3] a fast method is given for computing hyper-elliptic integrals. From [7] we know that an algebraic curve with genus $g = 1$ is birationally equivalent (i.e. the corresponding function fields are isomorphic) with a curve of the form $y^2 + x^3 + ax + b$. An integration problem over a curve in this normal form can be handled efficiently; see the computation times in [3], p. 214. This paper considers two problems. The first is how to implement the method in [7], Ch. 4, Proposition 4.6 on a computer. This gives the curve in normal form and an isomorphism of the function fields. The second problem is how to compute the inverse of this isomorphism. Afterwards we give some examples where this isomorphism can be applied to speed up the integration of algebraic functions with genus 1.

Our algorithm for computing this isomorphism of function fields is implemented in Maple 5.3 in the file `IntBasis`. The name of the procedure is `genus1`. A help text is included. Note that the Maple share library contains an older version of `IntBasis` which does not contain the procedure `genus1`. A more recent version is available via WWW at `http://www-math.sci.kun.nl/math/compalg/IntBasis` and by e-mail request.

The following can also be obtained from this WWW address: the papers [8, 9], more examples, comments on the implementation and a short description of different approaches that were implemented to test which was fastest.

2 Notations

- L is a field with characteristic 0. In the implementation L is the coefficients field of f, i.e. a field generated by the coefficients of f.
- f is an element of $L[x, y]$ which is irreducible in $\overline{L}[x, y]$.
- n is the degree of f as a polynomial in y.
- C is the projective algebraic curve given by f. We assume that the genus is equal to 1. This can be checked on a computer using Puiseux expansions, cf. section 3.1. A curve with genus 1 is called an *elliptic curve.*
- After applying a linear transformation we may assume that the point $(0, 1, 0)$ is not a point of C. This means that the degree of the curve is equal to n. We need this assumption to be able to apply the method in [9].
- $\overline{L}(C)$ is the function field of the curve. This field is identified with $\overline{L}(x)[y]/(f)$. Elements of this field are denoted as polynomials in y of degree $< n$.
- To avoid confusion we will use the variables x_0 and y_0 instead of x and y for denoting the normal form. The normal form f_0 is equal to $y_0^2 + x_0^3 + ax_0 + b$ for certain $a, b \in \overline{L}$. The polynomial f_0 describes a curve C_0. The function field $\overline{L}(C_0)$ is identified with $\overline{L}(x_0)[y_0]/(f_0)$. Elements of this field are denoted as polynomials in y_0 of degree < 2.
- The isomorphism $\theta : \overline{L}(C_0) \to \overline{L}(C)$ is determined by computing $\theta(x_0)$ and $\theta(y_0)$. To determine the inverse isomorphism we compute $\theta^{-1}(x)$ and $\theta^{-1}(y)$.

The usual definition of the Weierstrass normal form is $y^2 = 4x^3 + g_2x + g_3$ instead of the form $y^2 + x^3 + ax + b$ that we compute in this paper. We can call both Weierstrass normal forms because conversion is easy.

3 Computing θ and f_0.

The following construction is obtained from the proof of Proposition 4.6, Chapter 4 in Hartshorne's book.

- Compute a regular point p on the curve.
- Compute a function $\theta(x_0) \in \overline{L}(C)$ that has a pole of order 2 in p and no other poles.

- Compute a function $\theta(y_0) \in \overline{L}(C)$ that has a pole of order 3 in p and no other poles. Note that $\theta(x_0)$ and $\theta(y_0)$ are computed before θ and f_0 are known.

- Now $\theta(x_0)$ and $\theta(y_0)$ generate the function field, i.e. $\overline{L}(\theta(x_0), \theta(y_0)) = \overline{L}(C)$, and satisfy a relation $f_0 = y_0^2 + a_1y_0 + a_2x_0y_0 + a_3x_0^3 + a_4x_0^2 + a_5x_0 + a_6$. That means $\theta(f_0) = \theta(y_0)^2 + a_1\theta(y_0) + a_2\theta(x_0)\theta(y_0) + a_3\theta(x_0)^3 + a_4\theta(x_0)^2 + a_5\theta(x_0) + a_6 = 0$ in $\overline{L}(C)$. Now an isomorphism $\theta : \overline{L}(C_0) \to \overline{L}(C)$ is determined because θ is determined by the images of x_0 and y_0.

- After applying a linear transformation on $\theta(x_0)$ and $\theta(y_0)$ we may assume that f_0 is of the form $y_0^2 + x_0^3 + ax_0 + b$.

The next subsections describe how these steps can be implemented on a computer.

3.1 Computing the genus

$f \in L[x, y]$ describes a curve C in the projective plane P^2, although f allows us to "see" only the affine part of C. To view the whole curve C we must make f homogeneous. Let $F \in \overline{L}[x, y, z]$ be homogeneous of degree n such that $f = F_{z=1}$ (F with $z = 1$ substituted). Assume that the point $(0, 1, 0)$ is not a point on the curve. Let $P^1 \subset P^2$ be the line $y = 0$ (i.e. the set of points $(x, 0, z)$). Then C can be projected to P^1 as follows $(x, y, z) \in C \to (x, 0, z) \in P^1$. P^1 is identified with the set $\overline{L} \bigcup \{\infty\}$ by the map $(x, 0, z) \to x/z \in \overline{L} \bigcup \{\infty\}$. For $\alpha \in P^1$ the line $x = \alpha$ is the set of points $\{(\alpha, y, 1) | y \in \overline{L}\}$ if $\alpha \in \overline{L}$ and $\{(1, y, 0) | y \in \overline{L}\}$ if $\alpha = \infty$.

The Hurwitz formula (Corollary 2.4 in Hartshorne's book) gives a relation between the ramification indices e_P, the genus g of f and the genus of P^1 which is 0. Then the following formula is obtained

$$g = -n + 1 + \frac{1}{2} \sum_P (e_P - 1).$$

Consider the discriminant of F

$$d = \text{Res}_y(F, \frac{dF}{dy}).$$

This is a homogeneous polynomial in x and z of degree $n(n-1)$. Let $m_{(x,z)}$ be the multiplicity of the root (x, z) of d. The sum of $m_{(x,z)}$ taken over all $(x, z) \in P^1$ is $n(n-1)$. We compute $m_{(x,z)}$ as follows: $m_{(\alpha,1)}$ where $\alpha \in \overline{L}$ is the multiplicity of the root α in the polynomial $d_{z=1} = \text{Res}_y(f, \frac{df}{dy})$. $m_{(1,0)} = n(n-1) - \text{degree}_x(d_{z=1})$.

Let $p_1, \ldots, p_n \in \overline{L}[[x^{1/e}]]$ (for some $e \in \mathbb{N}$) be the Puiseux expansions of f at $x = 0$. Then

$$f = \prod_i (y - p_i)$$

where we assumed that f is monic as a polynomial in y. Let

$$\mathrm{Int}_{p_i} = \sum_{j \neq i} v(p_i - p_j)$$

where $v : \overline{L}[[x^{1/e}]] \to \frac{1}{e}\mathbb{N} \bigcup \{\infty\}$ is the valuation map $v(\sum a_i x^i) = \min_{a_i \neq 0} i$. Let e_{p_i} be the ramification index of p_i, i.e. the smallest integer such that $p_i \in \overline{L}[[x^{1/e_{p_i}}]]$. The multiplicity $m_{x=0}$ of the factor x in the discriminant of f is $\sum \mathrm{Int}_{p_i}$. The Puiseux expansions of f at $x = \alpha$ for $\alpha \in P^1$ can be computed by applying a transformation. If $\alpha = \infty$ we need to make f homogeneous to obtain F, then substitute $x = 1$ and take the line $z = 0$. The $m_{x=\alpha}$ for $\alpha \in P^1$ that is obtained this way is the same as the m_α defined before. The sum of m_α taken over all $\alpha \in P^1$ is $n(n-1)$ so the following formula is obtained

$$g = (n-1)(n-2)/2 - \frac{1}{2}\sum m_\alpha + \frac{1}{2}\sum(e_P - 1).$$

Define

$$\delta_{x=0} = \frac{1}{2}\sum_{i=1}^{n}(\mathrm{Int}_{p_i} - \frac{e_{p_i} - 1}{e_{p_i}}).$$

Similarly $\delta_{x=\alpha}$ for $\alpha \in P^1$ can be defined using the Puiseux expansions at $x = \alpha$. The sum of the Int_{p_i} equals $m_{x=0}$. The ramification index e_p of a Puiseux expansion is equal to the ramification index e_P of the corresponding place. Every place P corresponds to precisely e_P Puiseux expansions, hence the sum of $(e_p - 1)/e_p$ taken over these Puiseux expansions is $e_P - 1$. So the following formula holds

$$g = (n-1)(n-2)/2 - \sum_{\alpha \in P^1} \delta_{x=\alpha}.$$

Now $\delta_{x=\alpha} \neq 0$ if and only if there is a singularity on the line $x = \alpha$. In such a case the multiplicity of the root α of d is > 1. To find all singularities of f we must check all roots of d in P^1 with multiplicity > 1. If two roots α_1, α_2 are algebraically conjugated over the coefficients field of f then $m_{x=\alpha_1} = m_{x=\alpha_2}$. Hence we only need to compute the roots of d up to conjugation. The same holds for the Puiseux expansions. Every Puiseux expansion p gives a contribution $-\frac{1}{2}(\mathrm{Int}_p - \frac{e_p - 1}{e_p})$ to the genus. If two Puiseux expansions $p_1, p_2 \in \overline{L}[[(x-\alpha)^{1/e}]]$ at $x = \alpha$ are algebraically conjugated over $L(\alpha)((x-\alpha))$ then their contribution to the genus is the same. Hence in every conjugacy class only 1 Puiseux expansion needs to be computed.

The Puiseux expansions will later in the algorithm be used for computing two integral basis', cf. [8] and section 3.1 in [9]. Like for the genus computation it is sufficient to have the Puiseux expansions up to conjugation.

If P is a point on the curve define δ_P as the sum of the $\frac{1}{2}(\mathrm{Int}_p - \frac{e_p - 1}{e_p})$ taken over the Puiseux expansions p that correspond to the point P. Then we can rewrite the formula as $g = (n-1)(n-2)/2 - \sum \delta_P$ where the sum is taken over all points P of C in P^2. Without proof we mention that this δ_P corresponds to the δ_P in exercise 1.8, Chapter IV of Hartshorne's book.

If $g > 1$ the algorithm in section 3.3 will fail. It will try to determine $\theta(x_0)$ by solving a set of linear equations. However, if $g > 1$ a function $\theta(x_0)$ with only 1 pole of multiplicity 2 does not exist. Hence the computed set of linear equations will have no solutions. If $g = 0$ then we can compute a parametrization instead of a Weierstrass normal form. So if $g \neq 1$ there is no need to continue the computation and the implementation will exit returning a warning message.

3.2 Computing a regular point on the curve

Finding a point defined over $\overline{L}$ on the curve is an easy problem. Intersect the curve with a randomly chosen line and compute an intersection point. However, finding a good point is a difficult problem. Here "good" means that the point is defined over a small algebraic extension over L. The only algebraic extension over L that appears in the output of the algorithm is the extension that is used to find a regular point on the curve.

For the case $g = 0$ there is a way to find a good point cf. [10]. For the case $g = 1$ I do not know such a method, so currently only a heuristic for finding a regular point is implemented. The implementation looks for points on lines $x = \alpha$ through singularities and also looks on the line at infinity. If no regular point in an algebraic extension of degree $< n$ was found then the implementation looks on the lines $x = 0$, $x = 1$ etc. Then it is likely that an algebraic extension of degree n will be used which slows down the algorithm.

After a regular point is computed we compute a function P_1 with a pole of multiplicity 1 in this point. If the point is a finite point $x = \alpha$, $y = \beta$ then the function

$$P_1 = \frac{f_{x=\alpha}}{(y-\beta)(x-\alpha)}$$

is such a function. Here $f_{x=\alpha}$ stands for f with α substituted for x. Note that $y - \beta$ divides $f_{x=\alpha}$ in $\overline{L}[y]$. P_1 has no other poles in the finite part of the plane. If we start with a regular point in infinity we compute a function P_1 such that P_1 has a pole of order 1 in that point and no other poles at infinity.

3.3 Computing $\theta(x_0)$ and $\theta(y_0)$

This step could be done by an $L(D)$ computation. Methods for this are known including implementations cf. [1, 6]. Our implementation uses the method described in [9] which is based on integral basis computation [8]. We compute P_1 as in the previous section. Then take $P = -P_1^2 + rP_1$ as the starting function for obtaining $\theta(x_0)$. We use the minus sign to obtain $a_3 = 1$ in the next subsection. The term rP_1 where r is a variable is needed because the residue of the function $\theta(x_0)$ is not yet known. Now construct a function Q with undetermined coefficients in the same way as in [9]. Then find linear equations for these undetermined coefficients and for the variable r. Solving these equations gives $\theta(x_0) = P + Q$.

In the same way we can compute $\theta(y_0)$ using the starting function $P_1^3 + rP_1$. However, the starting function $P_1\theta(x_0) + rP_1$ gives the same result and is usually more efficient.

Is it efficient to use an integral basis computation for determining $\theta(x_0)$ and $\theta(y_0)$? Lemma 1 shows that the integral basis can be obtained from $\theta(x_0)$ and $\theta(y_0)$ if these two functions have their pole in a point at infinity. Assuming that this step is computationally cheap we conclude that computing an integral basis is not significantly more difficult than computing $\theta(x_0)$ and $\theta(y_0)$. So we can not lose much efficiency by computing an integral basis.

Lemma 1 *Let $f \in \overline{L}[x,y]$ describe an irreducible algebraic curve C. Let p be a place at infinity (i.e. a place where x has a pole). Let O be the integral closure of $\overline{L}[x]$ in $\overline{L}(C)$.*

- *Suppose the genus $g = 0$ and v is an element of the $\overline{L}(C)$ with only one pole on the curve, of multiplicity 1, in the place p. Then $O = \overline{L}[x, v]$.*
- *Suppose $g = 1$ and suppose v_2 and v_3 have only one pole on the curve, with multiplicity 2 resp. 3, in the place p. Then $O = \overline{L}[x, v_2, v_3]$.*

Proof: Let $p_1, p_2, \ldots, p_r$ be the infinite places $\neq p$. O is the set of functions in $\overline{L}(C)$ with no poles in the finite places (i.e. only poles in $p_1, \ldots, p_r$ and p). Let $R \subset O$ be the set of functions with no poles except in p. In the case $g = 0$ we have $R = \overline{L}[v]$ and in the case $g = 1$ we have $R = \overline{L}[v_2, v_3]$. This equality for $g = 1$ is proven as follows (the case $g = 0$ is proven in the same way). Since $v_2, v_3 \in R$ we have $R \supset \overline{L}[v_2, v_3]$. Take an element $e \in R$. Because $\overline{L}[v_2, v_3]$ contains elements $v_2, v_3, v_2^2, v_2v_3, v_2^3, v_2^2v_3, \ldots$ with pole orders $2, 3, 4, \ldots$ we conclude that we can subtract an element $e' \in \overline{L}[v_2, v_3]$ from e such that $e - e'$ has pole order < 2 in p. Since $g = 1$ and $e - e'$ has no other poles $e - e'$ must be a constant and hence in $\overline{L}$. So $e \in R$ and $R = \overline{L}[v_2, v_3]$.

The functions v (in the case $g = 0$) v_2 and v_3 (in the case $g = 1$) and x have their poles at infinity. So they are elements of O and hence $R[x] \subset O$. To prove the lemma we need to show $O \subset R[x]$. Choose $a \in O$. We want to construct an element $b \in R$ and a positive integer d such that bx^d has the same pole orders as a in the places $p_1, \ldots, p_r$. We choose d such that a/x^d has no poles at $p_1, \ldots, p_r$. Now we choose $b \in \overline{L}(C)$ with the following requirements:

- b has only one pole, at the place p (with no restrictions on the pole order of b at p).
- b has roots with precisely the same multiplicities as a/x^d in the places $p_1, \ldots, p_r$. (b may have roots in other places as well, only in this finite number of places $p_1, \ldots, p_r$ the multiplicity of b is specified).

Using the Riemann-Roch theorem it is not difficult to prove that such a b exists. Now $b \in R$ because p is the only pole of b.

Using this bx^d we can reduce the pole orders of a at $p_1, \ldots, p_r$. For any constant c the pole orders of $a - cbx^d$ at $p_1, \ldots, p_r$ are $\leq$ the pole orders of a. For a suitable c we have inequality for at least one p_i. Continuing this process we can subtract an element in $cbx^d + c_2b_2x^{d_2} + c_3b_3x^{d_3} + \ldots \in R[x]$ from a such that there are no poles left in $p_1, \ldots, p_r$. Then the function that remains is an element of R so we can conclude $a \in R[x]$.

□

3.4 Computing $a_1, \ldots, a_6$

We know from Hartshorne's book that $\theta(x_0)$ and $\theta(y_0)$ generate the function field and satisfy a polynomial relation $\theta(y_0)^2+a_1\theta(y_0)+a_2\theta(x_0)\theta(y_0)+a_3\theta(x_0)^3+a_4\theta(x_0)^2+a_5\theta(x_0)+a_6=0$. From this relation linear equations in a_i can be obtained. Solving these equations gives the polynomial relation $f_0 = x_0^2+a_1y_0+a_2x_0y_0+a_3x_0^3+a_4x_0^2+a_5x_0+a_6$ between $\theta(x_0)$ and $\theta(y_0)$. Then we get an isomorphism

$$\theta : \overline{L}(x_0)[y_0]/(f_0) \to \overline{L}(C).$$

The starting functions for computing $\theta(x_0)$ and $\theta(y_0)$ were chosen in such a way that $a_3 = 1$. The computation of linear equations for a_i can be speeded up by first substituting values for x (avoiding roots of denominators of $\theta(x_0)$ and $\theta(y_0)$) before the expression $\theta(y_0)^2+a_1\theta(y_0)+a_2\theta(x_0)\theta(y_0)+a_3\theta(x_0)^3+a_4\theta(x_0)^2+a_5\theta(x_0)+a_6$ is computed.

We can change $\theta(x_0)$ and $\theta(y_0)$ by linear transformations to obtain a new f_0 and θ in such a way that f_0 is of the form $y_0^2+x_0^3+ax_0+b$ cf. [7]. We skip the last normalization step $y_0^2+x_0(x_0-1)(x_0-\lambda)$ in [7] because that would introduce another (one extension over the coefficients field L has been made to find a point on the curve) algebraic extension in our algorithm.

4 Computing the inverse isomorphism.

4.1 Computing the minimum polynomial of x over $\overline{L}(\theta(x_0))$.

We start with a few facts about characteristic polynomials. The *characteristic polynomial* of an element $a \in L_2$ over a finite field extension $L_1 \subset L_2$ is defined as the characteristic polynomial of the L_1 linear map $L_2 \to L_2$ defined by the multiplication by a. Suppose $L_2 = L_1[y]/(f)$ where f is a monic polynomial in y. Then the characteristic polynomial of a is $r = \mathrm{Res}_y(t-a, f)$ (cf. [4],p. 162) where t is used as a variable for r. We have

$$r = m^{[L_2:L_1(a)]}$$

where m is the minimum polynomial of a over L_1. So m can be found by a resultant computation and a square free factorization.

Because $[\overline{L}(x,y) : \overline{L}(\theta(x_0))] = 2$ the number $e = [\overline{L}(x,y) : \overline{L}(\theta(x_0), x)]$ is 1 or 2. Now the characteristic polynomial r of $\theta(x_0)$ over $\overline{L}(x)$ can be computed as described (here $L_1 = \overline{L}(x)$ and $L_2 = \overline{L}(x,y)$). Then the minimum polynomial m of $\theta(x_0)$ over $\overline{L}(x)$ is found by a square free factorization of r. It is of degree $d = n/e$ ($n = \mathrm{degree}_y(f) = [\overline{L}(x,y) : \overline{L}(x)]$). Write m in the form

$$m = t^d + \frac{a_{d-1}}{a_d}t^{d-1} + \ldots + \frac{a_0}{a_d}t^0$$

for polynomials $a_i \in \overline{L}[x]$ with a_d of minimal degree. Note that $m^e = r$.

After multiplying with the denominator a_d we obtain the algebraic dependence $a_d\theta(x_0)^d + a_{d-1}\theta(x_0)^{d-1} + \ldots + a_0\theta(x_0)^0$ between $\theta(x_0)$ and x in $\overline{L}(x, \theta(x_0))$. Because the expression $\theta(x_0)$ contains the variable x we will write this expression as $\mu = a_d x_0^d + a_{d-1} x_0^{d-1} + \ldots + a_0 x_0^0$. Then we can view μ as a polynomial in x. It is the minimum polynomial of x over $\overline{L}(x_0)$. Here $\overline{L}(x_0)$ is identified with $\overline{L}(\theta(x_0))$. The degree of μ in x is $2/e$ because $[\overline{L}(\theta(x_0), x) : \overline{L}(\theta(x_0))] = [\overline{L}(\theta(x_0), \theta(y_0)) : \overline{L}(\theta(x_0))]/[\overline{L}(\theta(x_0), \theta(y_0)) : \overline{L}(\theta(x_0), x)] = 2/e$ (note that $\overline{L}(\theta(x_0), \theta(y_0)) = \overline{L}(x, y)$). So the degrees of a_i are $\leq 2/e$. Since $r = m^e$ we conclude that r is of the form

$$r = t^n + \frac{b_{n-1}}{b_n} t^{n-1} + \ldots + \frac{b_0}{b_n} t^0$$

for polynomials $b_i \in \overline{L}[x]$ of degree ≤ 2.

Using the fact that r has this form we can speed up the resultant computation $r = \mathrm{Res}_y(t - \theta(x_0), f)$. Before this resultant is computed we substitute a generic (values for which the denominator of $\theta(x_0) \in \overline{L}(x)[y]/(f)$ vanishes must be avoided) number $i \in L$ for x in $\theta(x_0)$ and f. This way $r_{x=i}$ (r with $x = i$ substituted) is obtained. After 5 of these resultant computations the coefficients of the polynomials b_i can be computed by solving linear equations. These equations must give a unique solution for r. To see this take two different rational functions k_1 and k_2, of which the degrees of the numerator and denominator are ≤ 2. If k_1 and k_2 take the same value for 5 different generic (i.e. not a pole of k_1 and k_2) values for x then $k_1 - k_2$ has 5 roots. Then $k_1 - k_2 = 0$ because the numerator of $k_1 - k_2$ is of degree ≤ 4.

4.2 Computing $\theta^{-1}(x)$

First determine the minimum polynomial (called m in the previous section) of $\theta(x_0)$ over $\overline{L}(x)$. After multiplying out the denominator the minimum polynomial of x over $\overline{L}(x_0)$ (called μ in the previous subsection) is obtained. If the degree in x of μ is 1 we can find $\theta^{-1}(x)$ in $\overline{L}(x_0)$ as a root of μ. Now assume the degree is 2. Then μ has 2 roots in $\overline{L}(C_0) = \overline{L}(x_0)[y_0]/(f_0)$. One of these is $\theta^{-1}(x)$. Compute both roots. Then we can check which one is $\theta^{-1}(x)$ by applying θ (this is done by substituting $\theta(x_0)$ for x_0 and $\theta(y_0)$ for y_0). This check is speeded up by first substituting a value in L for x.

The roots of μ in $\overline{L}(C_0)$ are computed as follows. We can write these roots as $c_0 + c_1 y_0$ and $c_0 - c_1 y_0$ with $c_0, c_1 \in \overline{L}(x_0)$. If $\mu = d_2 x^2 + d_1 x + d_0$ then $c_0 = -d_1/(2d_2)$. Now $c_1 y_0$ and $-c_1 y_0$ are roots of $\mu_{x=x+c_0} = d_2 x^2 - d_1^2/(4d_2) + d_0$. In $\overline{L}(x_0)[y_0]/(f_0)$ y_0^2 equals $-(x_0^3 + a x_0 + b)$ so c_1 and $-c_1$ are roots of $x^2 + (d_1^2/(4d_2^2) - d_0/d_2)/(x_0^3 + a x_0 + b)$. Now $c_1 \in \overline{L}(x_0)$ is found as a root of this polynomial.

In a similar way $\theta^{-1}(y)$ can be computed using the minimum polynomial of y over $\overline{L}(\theta(x_0))$.

5 A few examples

5.1 A sample computation

Consider the following example:

$$f = y^4 + xy^3 - x^2 + xy.$$

$(0,1,0)$ is not a point on the curve, so no linear transformation is needed. First find the singularities by factorizing the discriminant of f

$$\mathrm{Res}_y(f, \frac{df}{dy}) = -x^4(27x^4 + 446x^2 + 27).$$

Only at $\alpha \in \{0, \infty\}$ is $m_{x=\alpha} > 1$ (in both cases they are 4). Compute Puiseux expansions at $x = 0$: $p_1 = x + \ldots$, $p_2 = -x^{1/3} + \ldots$, p_3 is a conjugate of p_2 over $Q((x))$ and p_4 the other conjugate. The dots stand for terms with higher exponents in x. The contribution to the genus for p_1 is $\frac{1}{2}(1/3 + 1/3 + 1/3)$, the contribution of each p_2, p_3, p_4 is $\frac{1}{2}(1/3 + 1/3 + 1/3 - (3-1)/3)$ so the total contribution $\delta_{x=0} = 1$.

Compute Puiseux expansions at $x = \infty$. First make f homogeneous: $F = y^4 + y^3x + yz^2x - z^2x^2$, $F_{x=1} = y^4 + y^3 + yz^2 - z^2$. The Puiseux expansions at $z = 0$ are: $p_1 = -1 + \ldots$, $p_2 = z^{2/3} + \ldots$, p_3 is a conjugate of p_2 over $Q((z))$ and p_4 is the other conjugate. The contribution to the genus is 1. So the genus is 3(3-1)/2-1-1=1.

Now compute the integral closure of $\overline{Q}[x]$ in the function field. A basis as a $\overline{Q}[x]$ module is $[1, y, y^2, y^3/x]$. The integral closure of $\overline{Q}[[z]]$ in $\overline{Q}((z))[y]/(F_{x=1})$ is given by the $\overline{Q}[[z]]$ basis $[1, y, y^2, y^3/z]$.

Find a point on the curve: during the Puiseux computation we got lucky because the Puiseux expansions at $x = \infty$ show that $(-1, 1, 0)$ is a regular point. The function $P_1 = y^3/z$ in $\overline{L}(C) = \overline{Q}(z)[y]/(F_{x=1}) = \overline{Q}(x)[y]/(f)$ has a pole in this point. In $\overline{Q}(x)[y]/(f)$ syntax this is $P_1 = y^3/x^2$. The starting function $P = -P_1^2 - rP_1 = -x^3 + 2x^2y - 2xy^2 + rxy^3 + y^3 + x^2y^3)/x^3$ has only 1 pole at infinity, of multiplicity 2. Now we must add a function Q to P to eliminate the poles of P in the finite part of the projective plane. Given the bounds for the numerator and denominator in section 3.1 in [9] we can write

$$Q = \frac{1}{x^3} \sum_{j=0}^{3} \sum_{i=0}^{5-j} a_{ij} x^i y^j.$$

Now $P + Q$ must be an element of $\overline{Q}[x]1 + \overline{Q}[x]y + \overline{Q}[x]y^2 + \overline{Q}[x]y^3/x$. Eliminate denominators: then $x^3(P + Q)$ must be zero modulo $\{x^3, x^3y, x^3y^2, x^2y^3\}$. Compute the remainder of $x^3(P + Q)$ by reducing it with $\overline{Q}[x]$ multiples of the elements of $\{x^3, x^3y, x^3y^2, x^2y^3\}$. Equate the coefficients of this remainder (viewed as a polynomial in x and y) to zero. This results in the following set of equations $\{a_{00} = 0, a_{10} = 0, a_{20} = 0, a_{01} = 0, a_{11} = 0, a_{02} = 0, a_{22} = 0, r + a_{13} = 0, a_{12} - 2 = 0, 1 + a_{03} = 0, a_{21} + 2 = 0, a_{30} = 0\}$. Q must have no poles at infinity, so Q (first make Q homogeneous using the variable z and then substitute $x = 1$) must be an element of $\overline{Q}[[z]][1, y, y^2, y^3/z]$. This gives additional linear equations in the a_{ij} and r. Solving the equations gives $P + Q = -(x - y^3)/x$, so $\theta(x_0) = -(x - y^3)/x$. In the same way compute $\theta(y_0) = -x + 2y + y^3$.

Now write

$$f_0 = y_0^2 + x_0^3 + a_1y_0 + a_2x_0y_0 + a_3 + a_4x_0 + a_5x_0^2.$$

Find linear equations for a_i by substituting $\theta(x_0)$ for x_0 and $\theta(y_0)$ for y_0. To avoid getting large expressions we substitute an integer for x. This integer can not be 0 because x appears in the denominator of $\theta(x_0)$, so we substitute 1 for x. Now reduce the result modulo $f_{x=1}$ to obtain $(a_1 - 4a_5 + a_4 + 10 - 6a_2)y^3 + (2a_2 + 2a_5 - 6)y^2 + (2a_1 + 6 - 6a_2 - 2a_5)y + 2a_5 + a_3 - a_4 + 4a_2 - a_1 - 4$. All coefficients (as a polynomial in y) should be zero. The resulting linear equations do not yet determine all a_i, so we compute more linear equations by substituting the integer 2 for x instead of 1. Including these extra equations we can find $f_0 = y_0^2 + x_0^3 + 2x_0 + 3x_0^2$. Now substitute $x_0 - 1$ for x_0 in f_0 and replace $\theta(x_0)$ by $\theta(x_0) + 1$ to simplify f_0. Then we obtain

$$f_0 = x_0^3 - x_0 + y_0^2, \quad \theta(x_0) = \frac{y^3}{x}, \quad \theta(y_0) = -x + 2y + y^3.$$

Now we must compute the minimum polynomial of $\theta(x_0)$ over $\overline{\mathbb{Q}}(x)$. For this we need the resultant $r = \mathrm{Res}_y(t - \theta(x_0), f)$. In larger examples this resultant computation can often be very time consuming. Computing $r_{x=i}$ where i is a small integer (take integers $i = 0, 1, \ldots$ skipping those values which are a root of the denominator of $\theta(x_0)$) is much more efficient because then we need 1 variable less in the resultant computation. Write and ansatz r' for r

$$r' = t^4 + \sum_{i=0}^{3} \frac{a_{0,i}x^0 + a_{1,i}x^1 + a_{2,i}x^2}{a_{0,4}x^0 + a_{1,4}x^1 + a_{2,4}x^2} t^i.$$

Every $r_{x=i}$ gives the equation $(r')_{x=i} = r_{x=i}$. After multiplying with the denominator $a_{0,4}x^0 + a_{1,4}x^1 + a_{2,4}x^2$ we obtain linear equations in the a_{ij}. As explained in section 4.1 these linear equations for 5 different values of i suffice to determine r. In this example $r = -x^2 + 3tx^2 - 3t^2x^2 + t^3x^2 + t + 3t^2 + 3t^3 + t^4$. So the minimum polynomial of x over $\overline{\mathbb{Q}}(x_0)$ is $-x^2 + 3x_0x^2 - 3x_0^2x^2 + x_0^3x^2 + x_0 + 3x_0^2 + 3x_0^3 + x_0^4$. Make this polynomial monic: $x^2 + x_0(1 + 3x_0 + 3x_0^2 + x_0^3)/(-1 + 3x_0 - 3x_0^2 + x_0^3)$. The roots of this polynomial in $\overline{\mathbb{Q}}(x_0, y_0)$ are of the form y_0c and $-y_0c$ where $c \in \overline{\mathbb{Q}}(x_0)$ is a root of $x^2 + x_0(1 + 3x_0 + 3x_0^2 + x_0^3)/((-1 + 3x_0 - 3x_0^2 + x_0^3)(-x_0^3 + x_0))$. Factorization of this polynomial over $\overline{\mathbb{Q}}(x_0)[x]$ (in fact we can take $\mathbb{Q}$ instead of $\overline{\mathbb{Q}}$ because there can not appear any algebraic extensions over $\mathbb{Q}$ in the result that do not already appear in $\{f, f_0, \theta(x_0), \theta(y_0)\}$) gives the solutions $c = +/- \ (x_0+1)/(x_0^2 - 2x_0 + 1)$. Now $\theta^{-1}(x) = +/- \ \ y_0(x_0 + 1)/(x_0^2 - 2x_0 + 1)$. Which of these two values is correct can be decided by checking if $\theta(\theta^{-1}(x)) - x = 0$. However, before normalizing (i.e. writing this as a polynomial in y of degree < 4) this expression we first substitute a generic integer for x to speed up the computation. This zero test fails for $-y_0(x_0+1)/(x_0^2 - 2x_0 + 1)$ hence $\theta^{-1}(x) = y_0(x_0+1)/(x_0^2 - 2x_0 + 1)$. In the same way $\theta^{-1}(y) = -y_0/(-1 + x_0)$ can be computed.

5.2 An application to the integration of algebraic functions

This section gives a few examples where the isomorphism can be used to integrate algebraic functions of genus 1. Examples which require logarithmic extensions to find the integral are chosen because these are the hardest ones. As a first example take $f = y^4 - x^3 + x^2$ and $L = \mathbb{Q}$. The function field is

$\overline{Q}(x)[y]/(f)$. In this function field y stands for $(x^3-x^2)^{1/4}$. Now we want to integrate the algebraic function

$$a = \frac{(x-2)y(y^2+yx+x^2)}{x^2(x-1)(x^2-x+1)}.$$

The computation `int(a,x)` does not terminate in Maple 5.3 due to a bug. Marc Rybowicz did a bug-fix for this problem. Using this fix Maple can integrate a. Maple uses the Risch-Trager algorithm for this. The computation took 755 seconds.

We now try to do the same integration using the isomorphism θ and θ^{-1}. The computation of this isomorphism costs 4.3 seconds. Here computing the isomorphism means computing f_0, $\theta(x_0)$, $\theta(y_0)$, $\theta^{-1}(x)$ and $\theta^{-1}(y)$. The result in this example is $f_0 = y_0^2+x_0^3+x_0$, $\theta(x_0) = -y^2/(x(x-1))$, $\theta(y_0) = y/(x-1)$, $\theta^{-1}(x) = (x_0^2+1)/x_0^2$ and $\theta^{-1}(y) = y_0/x_0^2$. Now we can apply θ by substituting $\theta(x)$ for x and $\theta(y)$ for y. In the same way we can apply θ^{-1} using substitution. Note that for complicated curves (not in this example) applying θ and θ^{-1} often costs much more time than computing the isomorphism θ and θ^{-1}. The reason is that y_0 can appear in the denominator after this substitution $\theta(x)$ for x and $\theta(y)$ for y. Then we must do a normalization to represent this as an element of $\overline{L}(x_0)[y_0]/(f_0)$. This requires divisions in this function field. It turns out that in general these divisions are very costly operations. Our algorithm for computing the isomorphism avoids such divisions but they are still used for applying the isomorphism. We see from

$$\int a\ dx = \theta(\int \theta^{-1}(a) d\theta^{-1}(x)) = \theta(\int \theta^{-1}(a)\frac{d\theta^{-1}(x)}{dx_0}dx_0$$

that θ^{-1} must be multiplied with the derivative w.r.t. x_0 of $\theta^{-1}(x)$ before the integration algorithm is called. So we compute

$$a_0 = \theta^{-1}(a)\frac{d\theta^{-1}(x)}{dx_0} = -2\frac{x_0^5 - x_0^4y_0 + x_0^3y_0 - x_0y_0 - x_0 + y_0}{(x_0^6+2x_0^4+2x_0^2+1)x_0}.$$

The integration `int(a0,x0)` takes 22.5 seconds. Using Bertrand's implementation this integration takes only 2.3 seconds. Applying θ and θ^{-1} took 1.1 seconds in this example and checking the result took 0.5 seconds. So the total computation time was 8.2 seconds which is almost 100 times faster than the Risch-Trager implementation in Maple.

A reason that it works so well in this example is that θ and θ^{-1} were not very complicated. In other examples the algorithm often introduces an algebraic extension to find a point on the curve. Then this extension appears in the isomorphism as well. If the polynomial f has many terms then the size of the expressions $\theta(x_0)$, $\theta(y_0)$, $\theta^{-1}(x)$ and $\theta^{-1}(y)$ is usually much larger. In this case the bottleneck is often the normalization (i.e. removing y_0 from the denominator) of $\theta^{-1}(a)\frac{d\theta^{-1}(x)}{dx_0}$. So our method for integrating algebraic functions with genus 1 does not always work so well. On the other hand, the complexity of the Risch-Trager algorithm is very large as well.

In the following example y is the algebraic function defined by the minimum polynomial $f = y^4 + x^3y - x^2$ and $a = (54x^7 + 27x^4y^3 - 27x^5y + 36x^3y^2 + 108x^4 + 144xy^3 + 48x^2y + 192y^2 + 320x)/((27x^6 + 256)x^2)$. Like in the previous example our heuristic for finding a point on the curve does not introduce an algebraic extension in this example. In this example our method combined with Bertrand's implementation uses 9.2 seconds to integrate a, where `int(a,x)` in Maple 5.3 + bug-fix takes 6400 seconds.

References

[1] D. Le Brigand, J.J. Risler, *Algorithme de Brill-Noether et codes de Goppa*, Bull. Soc. math. France, 116 231-253 (1988)

[2] D. G. Cantor, *Computing in the Jacobian of an Hyperelliptic Curve*, Math. of Comp. Vol. 48, No. 177, 95-101 (1987)

[3] L. Bertrand, *On the Implementation of a new Algorithm for the Computation of Hyperelliptic Integrals*, ISSAC '94 Proceedings, 211-215 (1994)

[4] H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer-Verlag (1993).

[5] Duval, D., (1989). *Rational Puiseux expansions*, Compos. Math. 70, No. 2, 119-154

[6] G. Haché, D. Le Brigand *Effective Construction of Algebraic Geometry Codes* Rapport de recherche INRIA, No 2267 (1994)

[7] R. Hartshorne, *Algebraic Geometry* Springer-Verlag (1977)

[8] M. van Hoeij, *An algorithm for computing an integral basis in an algebraic function field*, To appear in J. Symbolic Computation.

[9] M. van Hoeij, *Computing parametrizations of rational algebraic curves*, ISSAC '94 Proceedings, 187-190 (1994)

[10] J.R. Sendra, F. Winkler, *Determining Simple Points on Rational Algebraic Curves*, RISC-Linz Report Series No. 93-23 (1993)

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