# Computing the System of Adjoint Plane Curves by Puiseux Expansion

## Part III: Computational Complexity [1]

Peter Stadelmeyer

*Research Institute for Symbolic Computation*
*Johannes Kepler Universität*
*Linz, Austria*

Peter.Stadelmeyer@risc.uni-linz.ac.at

**Abstract**

Let $\mathcal{C} \subset \mathbf{A}^2_K$ be an irreducible algebraic curve in the affine plane. The main result of this paper is a detailed complexity analysis of the Newton–Puiseux algorithm used to resolve the singularities of $\mathcal{C}$. This analysis is then used to estimate the computational complexity for computing the linear system of adjoint curves of $\mathcal{C}$ of some given degree.

Contents



[1] Supported by the Austrian Fonds zur Förderung der wissenschaftlichen Forschung project number P11160-TEC "HySaX".

## 1. Introduction

This is the third and last part of a series of reports about computing the system of adjoint plane curves with the help of the Newton–Puiseux algorithm. Whereas the first two parts cover theoretic and algorithmic issues, this part deals with the computational complexity.

Let $\mathcal{C} \subset \mathbf{A}^2_K$ be an irreducible algebraic curve in the affine plane defined by $f(x,y) \in k[x,y]$ and let $d$ be some natural number. By $\mathrm{Adj}_d(f)$ we denote the linear system of curves of degree $d$ which are adjoint to $\mathcal{C}$.

Most methods we know for computing $\mathrm{Adj}_d(f)$ have in common that in a first step the singular points of $\mathcal{C}$ have to be resolved. In a second step a linear system of equations built from the resolution-information has to be solved. As we have shown in the previous parts there exist several methods to resolve curve singularities. So e.g. projective and affine quadratic transformations, toroidal transformations, and Newton transformations. The latter are well-known from the Newton–Puiseux algorithm for computing Puiseux-series expansions. In our opinion they are also a computationally very efficient method to compute the adjoints. Together with a definition of "being adjoint" using intersection multiplicities we are able to construct a linear system of equations describing the adjoint curves in a very natural way. Such an algorithm is described in [Stadelmeyer, 2000].

In the following we give a detailed complexity analysis for the key part of this algorithm, namely for resolving the singularities by Puiseux-series expansion. Once this is done, from the before mentioned it follows that a complexity analysis for computing the adjoint curves can then be easily obtained, since the complexity for solving a linear system is well-known.

One could say, that there are basically three quantities which contribute to the complexity of the resolution algorithm based on Puiseux-series expansion.

- The number of Newton transformations necessary to separate all curve-branches of a singularity, i.e. to compute the *singular part* of the Puiseux-series, see the paper of D. Duval [Duval, 1989].
- The $x$- and $y$-degrees of the transformed polynomials.
- The degrees of the algebraic extensions necessary to represent the coefficients of the Puiseux-series.

We do not consider the size of the coefficients occurring during computation. A bound for the coefficients of the Puiseux-series and a bit-complexity analysis was given by Walsh, see [Walsh, 1998] and [Walsh, 1999].

Let $M$ and $N$ be the $x$- and $y$-degree of $f(x,y)$. For each of the above three quantities there are rather obvious bounds, namely:

- The number of Newton transformations for computing the singular part of a Puiseux-series is bounded by $MN$.

- The $x$- and $y$-degrees of the transformed polynomials are bounded by $MN^2$ and $N$, respectively.
- The degrees of the algebraic extensions are at most $N$.

Putting these bounds together it follows that the singular parts of all curve-branches of a singularity can be computed with no more than $M^2N^6$ operations in the ground field $k$. This is the bound given by D. Duval.

But this complexity bound has two disadvantages: First, it is a *global bound*, which means that the complexity of the singularity, measured by some invariant as e.g. its Milnor-number, is neglected in the analysis. A complexity analysis based completely on local data was done by J. Teitelbaum. In his paper, see [Teitelbaum, 1990], he shows that at most $\mu^6$ operations are necessary, where $\mu$ is the Milnor-number. Second, it is not taken into consideration that not all three quantities can contribute to the complexity at the *same time*. If a resolution of a singularity yields an algebraic extension of high degree, the number of Newton transformations to compute its singular parts has to be small.

Taking all this into account and making extensive use of the connection between quadratic transformations (and the knowledge about their properties) and Newton transformations, we were able to show that resolving a plane curve singularity takes no more than $\mu^3$ operations in the ground field, where $\mu$ is again its Milnor-number.

The paper has the following structure. In Section 2 we clarify notation and briefly describe the rational Newton–Puiseux algorithm from D. Duval. Also, as an example, a family of curves is presented for which the number of Newton transformations to compute the singular part of a curve-branch is of order $MN$, i.e. it is shown that the before mentioned bound is optimal up to terms of lower order. In Section 3 we just present the idea of passive factorization as described by D. Duval. No new results are presented there. The main results of this paper are presented in Section 4, where a detailed complexity analysis of the Newton–Puiseux algorithm is done. As shown in Section 5 and Section 6 this can than be used to derive a complexity bound for resolving all singularities of a curve and for computing the system of adjoint curves.

We would like to explicitly thank Franz Winkler for spending many hours on discussing this topic. It was also he, who originally suggested this problem to us.

## 2. Newton–Puiseux Algorithm

Throughout this paper, let $k$ be a commutative field with characteristic 0, e.g. the rational numbers, and let $K$ be its algebraic closure. The algebraically closed field of Puiseux-series over $K$ in the indeterminate $t$ is written as $K\langle\langle t\rangle\rangle$.

Let $f(x,y) \in k[x,y]$ be an absolutely irreducible polynomial of $y$-degree $N$ and monic in $y$, i.e.

$$f(x,y) = y^N + a_{N-1}(x)y^{N-1} + \cdots + a_1(x)y + a_0(x),$$

where $a_i(x) \in k[x]$, for $0 \leq i < N$. Since we will also refer to the $x$-degree and total-degree of $f(x,y)$ we denote them by $M$ and $D$ respectively.

The polynomial $f(x,y)$ defines a plane *algebraic curve* $\mathcal{C}$ in the affine plane $\mathbf{A}_K^2$, namely the set of points given by

$$\mathcal{C} = \{(x_0, y_0) \in \mathbf{A}_K^2 \mid f(x_0, y_0) = 0\}.$$

An affine *curve-branch* $\mathfrak{z}$ *with origin* $q$ is defined as the set of points

$$\mathfrak{z} = \{(A(t), B(t)) \mid t \in K\},$$

where $A(t)$, $B(t)$ are power-series in $K[[t]]$ not both constant and $q = (q_x, q_y) = (A(0), B(0))$. We say that $\mathfrak{z}$ is a *curve-branch of* $\mathcal{C}$ *at* $q$ or $\mathfrak{z}$ is a *curve-branch of* $\mathcal{C}$ *over* $q_x$ if $f(A(t), B(t)) = 0$.

In the following we will describe the well-known Newton–Puiseux algorithm to compute the Puiseux-series $\mathfrak{B}_i(t)$, for $1 \leq i \leq N$, of $f(x,y)$ at $x = 0$. There are at least two different ways to interpret them. First, they can be seen from a function theoretic point of view as the *roots of* $f(x,y)$ *in* $K\langle\langle x \rangle\rangle$, i.e.

$$f(x,y) = \prod_{i=1}^{N} (y - \mathfrak{B}_i(t)).$$

This was the original motivation of Newton (see [Newton, 1660]) to introduce them. Second, from a geometric point of view they can be seen as *local parameterizations of* $\mathcal{C}$ *at* $x = 0$, i.e.

$$f(t, \mathfrak{B}(t)) = 0 \quad , \text{for } 1 \leq i \leq N.$$

For more details we refer to [Walker, 1978], [Brieskorn and Knörrer, 1986] or [Abhyankar, 1990].

2.1. **Algorithm.** In this subsection we give a brief description, just to clarify notation, of a rational version of the Newton–Puiseux algorithm as formulated by D. Duval. For details we refer to [Duval, 1989].

The two basic concepts of the Newton–Puiseux algorithm are Newton polygons and Newton transformations which we will define now. Details can again be found e.g. in [Walker, 1978], [Brieskorn and Knörrer, 1986].

**Definition 1.** Let $g(x,y) = \sum_{i=0}^{M} \sum_{j=0}^{N} a_{i,j} x^i y^j \in K[x,y]$. The *Newton-polygon* of $g(x,y)$ is defined as the lower part of the convex hull of the set of points $\{(i,j) \mid a_{i,j} \neq 0\}$ in the $(i,j)$-plane.

Let $E$ be the number of segments of the Newton-polygon. For each segment $S_e$, for $1 \leq e \leq E$, we define the numbers $p'_e$ and $q'_e$ as shown in Figure 1. Let $d'_e = \gcd(p'_e, q'_e)$ and $p_e = p'_e/d'_e$, $q_e = q'_e/d'_e$.

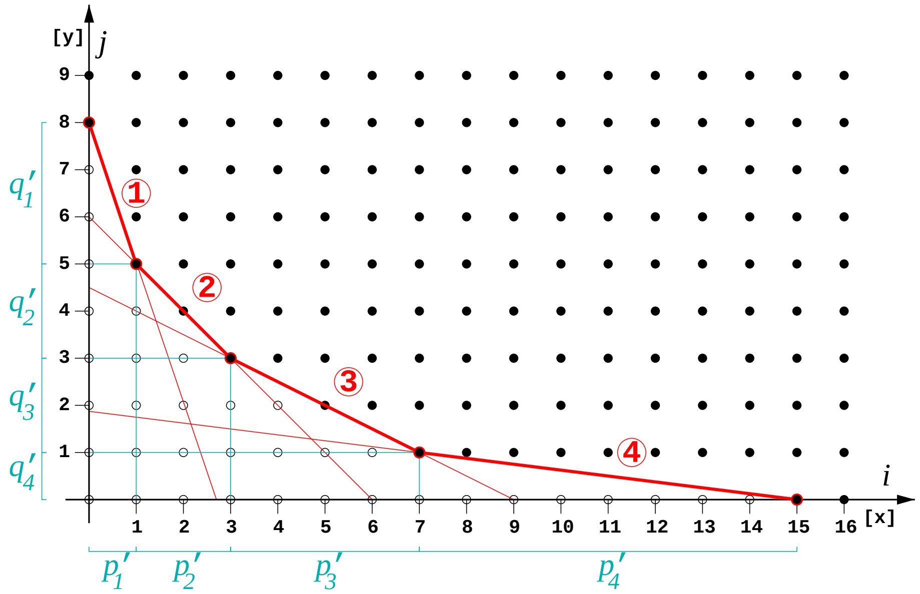


FIGURE 1. Newton Polygon

The line equation in the $(i, j)$-plane associated to $S_e$ is then defined by

$$L_e \colon q_e i + p_e j = l_e.$$

Finally, we define for each segment $S_e$ the numbers $a_e$ and $b_e$ as

$$a_e = \sum_{1 \le e' < e} p'_{e'},$$
$$b_e = \sum_{e < e' \le E} q'_{e'}.$$

As can easily be seen if holds that $l_e = a_e + p_e + \frac{p_e}{q_e} b_e$, for all $1 \le e \le E$.

**Definition 2.** A *Newton transformation (NT)* is defined as a 6-tuple $\tau = [p, q, l;\ \xi, u, v]$, where $p$, $q$, $l \in \mathbb{N}$, with $\gcd(p, q) = 1$, $\xi \in K$, and $u$, $v \in \mathbb{Z}$.

Let $g(x, y) \in K[x, y]$. We then define the *Newton transform of* $g(x, y)$ *w.r.t.* $\tau$, written as $\tau(g(x, y))$, as

$$\tau(g(x, y)) = x^{-l} g(\xi^{-v} x^q, x^p(\xi^u + y)).$$

We are now able to define one step in the Newton–Puiseux algorithm. Let $g(x, y) \in K[x, y]$ be a polynomial and $S$ a segment of its Newton polygon. We assume that $S$ is on the line defined by $qj + pi = l$ in the $(i, j)$-plane. We define $\Phi(z) \in K[z]$ as $\Phi(z) = \sum_{(i,j) \in S} a_{i,j} z^{(i-i_0)/q}$, where $i_0$ is the smallest value of $i$ such that there is a point $(i, j)$ on $S$. Let $\xi \in K$ be a root of $\Phi(z)$.

Since $\gcd(p, q) = 1$ there exist $u, v \in \mathbb{Z}$ such that $uq + vp = 1$. The NT (w.r.t. $S$ and $\xi$) is then defined as

$$\tau = [p, q, l;\ \xi, u, v],$$

and the transformed polynomial $g_{\mathrm{new}}(x, y)$ as

$$g_{\mathrm{new}}(x, y) = \tau(g(x, y)).$$

Note, that the integers $u$ and $v$ are not uniquely determined. Disregarding the requirement $uq + vp = 1$ and always setting $u = 1$, $v = 0$ yields the classical Newton–Puiseux algorithm.

**Example 1.** To illustrate the above definitions we consider the Newton-polygon of the polynomial

$$g(x, y) = y^8 - xy^5 + x^3y^3 + 2x^5y^2 + x^7y + x^{15}.$$

Its four segments and NTs are defined by the following table (see also Figure 1).

| $e$ | $p_e$ | $q_e$ | $l_e$ | $a_e$ | $b_e$ | $\Phi_e(z)$ | $u_e$ | $v_e$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 3 | 8/3 | 0 | 5 | $z - 1$ | 0 | 1 |
| 2 | 1 | 1 | 3 | 1 | 3 | $-z^2 + 1$ | 1 | 0 |
| 3 | 2 | 1 | 9/2 | 3 | 1 | $z^2 + 2z + 1$ | 1 | 0 |
| 4 | 8 | 1 | 15 | 7 | 0 | $z + 1$ | 1 | 0 |

**Definition 3.** The *singular part of a Puiseux-series* consists of all those terms which correspond to NTs $\tau = [p, q, l;\ \xi, u, v]$, where $\xi$ is a multiple root. By a *singular part of a curve-branch* $\mathfrak{z}$ we mean the singular part of the Puiseux-series of $\mathfrak{z}$.

A basic property of the rational Newton–Puiseux algorithm is, that after having computed all singular parts of $f(x, y)$ there is a one-to-one correspondence between the curve-branches at $x = 0$ and the Puiseux-series computed by the algorithm (see Theorem 3 of [Duval, 1989]).

Of course, there are infinitely many different rational Puiseux-series corresponding to one curve-branch (e.g. obtained by a linear parameter substitution). From now on, whenever we speak about *the* Puiseux-series of a curve-branch, we mean the one computed by the algorithm above.

From now on we assume that the curve-branches of $\mathcal{C}$ over $x = 0$ are given by $\mathfrak{z}_i$, for $1 \le i \le I = N$.

**Definition 4.** For each curve-branch $\mathfrak{z}_i$, for $1 \le i \le I$, of $\mathcal{C}$ over $0$ we define:

| | |
|---|---|
| $f_{i,k}(x, y)$ | The polynomial defined as $\tau_{i,k-1}(f_{i,k-1}(x, y))$, where $f_{i,0}(x, y) = f(x, y)$. |
| $S_{i,k}$ | The segment of the Newton-polygon of $f_{i,k}(x, y)$ belonging to $\mathfrak{z}_i$. |
| $p_{i,k}$, $q_{i,k}$, $l_{i,k}$, $a_{i,k}$, $b_{i,k}$ | Defined as in Definition 1. |
| $\tau_{i,k}$ | The NT defined by $f_{i,k}(x, y)$ and $S_{i,k}$. |
| $K_i$ | The number of NTs (i.e. the number of steps in the Newton–Puiseux algorithm) necessary to compute the singular part of $\mathfrak{z}_i$. |

It is a consequence of the one-to-one correspondence between curve-branches and rational Puiseux-series that the above polynomials, segments, numbers, and NTs are all well and uniquely defined.

2.2. **Deferred Expansion.** During the computation of the Puiseux-series we use *deferred expansions* for the intermediate polynomials $f_{i,k}(x, y)$. By this we do not mean lazy-evaluation as known from power-series computation, since the polynomials $f_{i,k}(x, y)$ contain at any time all their coefficients. We rather mean, that we allow a collected and nested representation of the $f_{i,k}(x, y)$'s for all terms with $x$-degree greater than or equal to some fixed number.

As we will see later-on, this allows us to make practical use of the theoretical $x$-degree bounds derived in Section 4. To make things precise we define the following.

**Definition 5.** Let $g(x, y) \in K[x, y]$ be a polynomial. We say that $g(x, y)$ is $x$-*expanded up to order* $l$ if $g(x, y)$ is of the form

$$g(x, y) = \sum_{\delta=0}^{\Delta} x^{\delta} g_{\delta}(x, y),$$

where $g_{\delta}(x, y) \in K[y]$, for $0 \leq \delta \leq l$, and $g_{\delta}(x, y) \in K[x, y]$, for $l < \delta \leq \Delta$.

Recalling the definition of a NT it is easily seen, that a polynomial which is $x$-expanded up to a high enough order is, after applying a NT to it, again $x$-expanded up to some order.

**Lemma 1.** *Let* $g(x, y) = \sum_{\delta=0}^{\Delta} x^{\delta} g_{\delta}(x, y) \in K[x, y]$ *be a polynomial and let* $\tau = (p, q, l;\ \xi, u, v)$ *be a NT derived from the Newton-polygon of* $g(x, y)$. *If* $g(x, y)$ *is* $x$*-expanded up to order* $l + k$, *then* $\tau(g(x, y))$ *is* $x$*-expanded up to order* $k$ *just by substitution.*

*Proof.* Let $g_\delta(x,y) = \sum_{j=0}^{n_\delta} c_{\delta,j} y^j$, for $0 \le \delta \le l+k$. It holds then that

$$\begin{aligned}\tau(g(x,y)) = \sum_{\delta=0}^{\Delta} \tau(x^\delta g_\delta(x,y)) = \\ \sum_{\delta=0}^{\Delta} \xi^{-\nu\delta} x^{q\delta-l} g_\delta(\xi^{-\nu}x^q, x^p(\xi^u+y)) = \\ \sum_{\delta=0}^{l+k} \xi^{-\nu\delta} \sum_{j=0}^{n_\delta} c_{\delta,j} x^{q\delta+pj-l}(\xi^u+y)^j + \\ \sum_{\delta=l+k+1}^{\delta} \xi^{-\nu\delta} x^{q\delta-l} g_\delta(\xi^{-\nu}x^q, x^p(\xi^u+y)).\end{aligned}$$

As we can see, all $x$-degrees in the last sum are greater than $k$ and hence no expansion is necessary to $x$-expand $\tau(g(x,y))$ up to order $k$. □

The next Lemma shows that it makes no difference if we compute the coefficients of the polynomials $f_{i,k}(x,y)$ at each step up to some given $x$-degree (i.e. full expansion up to some $x$-degree and truncating all higher $x$-degree terms) or if we use deferred expansions up to the same $x$-degrees. We will make use of this result later on when we count the number of operations necessary to resolve a singular point of $\mathcal{C}$, since in this situation we do not know the required $x$-degrees in advance, but they are determined by the Newton-polygons of the $f_{i,k}(x,y)$'s.

**Lemma 2.** *Let*

$$f_{i,k}(x,y) = \sum_{\delta=0}^{\Delta_{i,k}} x^\delta f_{i,k,\delta}(x,y),$$

*where* $f_{i,k,\delta}(x,y) \in k[x,y]$, *for* $0 \le \delta \le \Delta_{i,k}$. *It needs no more than*

$$\left(\sum_{k'=0}^{k-1}(k'+1)l_{i,k'} + k\delta_0\right) N$$

*multiplications to expand all* $f_{i,k,\delta}(x,y)$, *for* $\delta \le \delta_0$.

*Proof.* By induction.

- $k=0$:
  Since we assume that the input polynomial is given in expanded form nothing has to be shown.
- $k \to k+1$:
  To expand all $f_{i,k+1,\delta}(x,y)$, for $\delta \le \delta_0$ we have first to expand all $f_{i,k,\delta}(x,y)$, for $\delta \le l_{i,k} + \frac{\delta_0}{q_{i,k}}$. Using the induction hypothesis this

can be done needing no more than

$$\left(\sum_{k'=0}^{k-1}(k'+1)l_{i,k'}+k(l_{i,k}+\frac{\delta_0}{q_{i,k}})\right)N$$

multiplications. In addition to that $(l_{i,k}+\frac{\delta_0}{q_{i,k}})N$ multiplications are needed to expand all $f_{i,k+1,\delta}(x,y)$, for $\delta \le \delta_0$, once the $f_{i,k,\delta}(x,y)$, for $\delta \le l_{i,k}+\frac{\delta_0}{q_{i,k}}$, are already expanded. So we get in total

$$\left(\sum_{k'=0}^{k-1}(k'+1)l_{i,k'}+k(l_{i,k}+\frac{\delta_0}{q_{i,k}})+l_{i,k}+\frac{\delta_0}{q_{i,k}}\right)N=$$
$$\left(\sum_{k'=0}^{k}(k'+1)l_{i,k'}+(k+1)\frac{\delta_0}{q_{i,k}}\right)N\le\left(\sum_{k'=0}^{k}(k'+1)l_{i,k'}+(k+1)\delta_0\right)N.$$

□

2.3. **Maximal Number of Newton Transformations.** In this subsection we deal with the question of how many NTs it may take at most to compute the singular part of a curve-branch. We give two upper bounds, one of global and one of local character and show that both are of optimal order.

**Lemma 3.** *The number* $K$ *of NTs necessary to compute the singular part of the Puiseux-series of a curve-branch* $\mathfrak{z}$ *of* $f(x,y)$ *is bounded by*

$$K \le 2(N-1)M+1 \tag{1}$$

*or*

$$K \le \mu, \tag{2}$$

*where* $\mu$ *is the Milnor-number of* $\mathfrak{z}$.

*Proof.* The proof of (1) is given in [Duval, 1989], whereas inequality (2) follows from the correspondence of NTs and QTs. Each NT in the singular part of the algorithm stands for at least one QT with a neighboring point of multiplicity greater than 1. See [Stadelmeyer, 2000] for further details. □

*Remark* 1. Also a degree bound for the singular part of a curve-branch of $f(x,y)$ can be found in the paper of Duval, see page 151. The singular part is at most of degree $2(N-1)M$.

*Remark* 2. A direct consequence of (2) in Lemma 3 is that $K$ also bounded by $(D-1)(D-2)$.

We will now show, that these bounds are up to multiplicative constants and terms of smaller order optimal. To do so, we define two families of polynomials ($n,\ k \in \mathbb{N}$):

$$\begin{aligned} F_{n,k}(x,y) &= y^{4n+2} + 2xy^{2n+1} + x^2 + x^{2k+3} \\ G_n(x,y) &= F_{n,2n-1}(x,y) \end{aligned}$$

The polynomials (respectively the curves they define) have the following properties:

(i) The degrees of $F_{n,k}(x,y)$ and $G_n(x,y)$ are given by:

| | M ($x$-degree) | N ($x$-degree) | D (total-degree) |
|---|---|---|---|
| $F_{n,k}(x,y)$ | $2k+3$ | $4n+2$ | $\max(2k+3, 4n+2)$ |
| $G_n(x,y)$ | $4n+1$ | $4n+2$ | $4n+2$ |

(ii) The Newton-transformations for $F_{n,k}(x,y)$ and $G_n(x,y)$ are

$$\begin{aligned} \tau_1 &= [1, 2n+1, 4n+2;\ -1, 0, 1], \\ \tau_2 &= [(2n+1)(2k+1), 2, (4n+2)(2k+1); \\ &\qquad (2n+1)^{-2}, -(2nk+n+k), 1], \end{aligned}$$

and

$$\begin{aligned} \tau_1 &= [1, 2n+1, 4n+2;\ -1, 0, 1], \\ \tau_2 &= [8n^2+2n-1, 2, 16n^2+4n-2; \\ &\qquad (2n+1)^{-2}, -(4n^2+n-1), 1]. \end{aligned}$$

(iii) Each of the curves defined by $F_{n,k}(x,y)$ and $G_n(x,y)$ has at $(0,0)$ one curve-branch whose generalized Puiseux-parameterization is

$$\begin{aligned} F_{n,k}(x,y): x(t) &= -(2n+1)^{4n+2}t^{4n+2}, \\ y(t) &= (2n+2)^2t^2 + (2n+1)^{(2n+1)(2k+1)+1}t^{(2n+1)(2k+1)+2}, \end{aligned}$$

and

$$\begin{aligned} G_n(x,y): x(t) &= -(2n+1)^{4n+2}t^{4n+2}, \\ y(t) &= (2n+2)^2t^2 + (2n+1)^{2n(4n+1)}t^{2n(4n+1)+1}. \end{aligned}$$

(iv) The proper multiplicity sequences of the above curve-branches are

$$\begin{aligned} F_{n,k}(x,y) &: [4n+2]^0[2]^{2n+1}[2]^{(2k+1)n+k}[1]^2, \\ G_n(x,y) &: [4n+2]^0[2]^{2n+1}[2]^{n(4n+1)-1}[1]^2, \end{aligned}$$

and the free and satellite neighboring points are given by

$$\begin{aligned} F_{n,k}(x,y) &: [\mathrm{F}][\mathrm{S}]^{2n}[\mathrm{F}]^{(2k+1)n+k+1}[\mathrm{S}][\mathrm{F}], \\ G_n(x,y) &: [\mathrm{F}][\mathrm{S}]^{2n}[\mathrm{F}]^{n(4n+1)}[\mathrm{S}][\mathrm{F}]. \end{aligned}$$

(v) The Milnor-numbers of the singularities at $(0,0)$ of $F_{n,k}(x,y)$ and $G_n(x,y)$ are

$$F_{n,k}(x,y) : \mu = (2n+1)k + 3n + 1,$$
$$G_n(x,y) : \mu = 4n^2 + 3n.$$

(vi) The curve defined by $G_n(x,y)$ has only one singularity, namely $(0,0)$, and its genus is

$$g = \frac{(4n+1)4n}{2} - (4n^2 + 3n) = 4n^2 - n.$$

If we now add some "suitable" terms of "suitable" high degree to $F_{n,k}(x,y)$ and $G_n(x,y)$ we can achieve that the free neighboring points get coordinates unequal to zero. This way we get non-zero coefficients for the Puiseux-series and the number of NTs is equal to the number of free neighboring points. For more details we again have to refer to [Stadelmeyer, 2000].

A suitable modification term is

$$H_n(x,y) = (4n+2)(x^2y - xy^{2n+2}) + (2n+1)^2x^2y^2$$

and the modified polynomials are

$$F^*_{n,k}(x,y) = F_{n,k}(x,y) + H_n(x,y),$$
$$G^*_n(x,y) = G_n(x,y) + H_n(x,y).$$

So to summarize, we have constructed a family of curves with increasing genus where each curve has a singularity at $(0,0)$. Further, the number of NTs necessary to resolve the singularity at $(0,0)$ is of order $MN$ or $\mu$.

## 3. Computing with Algebraic Numbers

As we saw in the last Section the coefficients of a Puiseux-series are in general algebraic numbers (roots of a univariate polynomial defined by the Newton-polygon). Therefore, when computing the singular part of a curve-branch we have to deal with multiple algebraic extensions of the ground field.

There are at least two different ways to handle these multiple algebraic extensions: Either by repeatedly computing primitive elements or by introducing a new variable for each algebraic extension.

3.1. **Primitive Elements.** A classical algorithm for computing a primitive element of two algebraic numbers is described in [van der Waerden, 1991] or [Winkler, 1996]. There one has to compute the gcd of $g(x)$ and $p(x)$ in $k(\gamma)$, where $\gamma$ is an algebraic number of degree $d$ and $\deg(g)\deg(p) = d$. It is known that this gcd-computation can be done in $d^3$ $k$-operations. As far as we know, this is also a lower bound for all currently known algorithms to compute a primitive element.

3.2. **Multiple Variables.** Another possibility to compute in $k(\alpha_1, \dots, \alpha_n)$ is to introduce for each $\alpha_i$, for $1 \leq i \leq n$, a new variable and a minimal polynomial $p_i(x) \in k(\alpha_1, \dots, \alpha_{i-1})[x]$. Since $k(\alpha_1, \dots, \alpha_i)$ is isomorphic to $k(\alpha_1, \dots, \alpha_{i-1})[x]/p_i(x)$, we are able to compute in $k(\alpha_1, \dots, \alpha_n)$. Let $d$ be the degree of the algebraic extension, i.e. $d = [k(\alpha_1, \dots, \alpha_n) : k]$. Each operation in $k(\alpha_1, \dots, \alpha_n)$ can then be replaced by at most $d^2$ operations in $k$.

As explained by D. Duval the above idea can be extended to the case where the polynomials $p_i$ are no longer necessarily prime but just square-free. We therefore do not compute in a field, but in a ring with zero divisors. Since square-freeness can be checked by gcd-computation, polynomial factorization can be avoided at all. For further details about this method, called the D5-method, we refer to [Dicrescenzo and Duval, 1984], [Dicrescenzo and Duval, 1985], or [Duval, 1994].

What we have to take additionally into account from now on are zero-equality tests. But if a term $a(x)$ contains the zero element can again be decided by gcd-computation. If the gcd is non-trivial, we have to consider two different cases. At the same moment we also found a non-trivial factor of some $p_i(x)$. For this reason, the D5-method is also called the method of passive factorization.

We assume that at every step in our algorithm all minimal polynomials $p_i(x)$, $1 \leq i \leq n$, are monic and fully reduced w.r.t. $p_{i'}(x)$, for $1 \leq i' < i$.

A typical case splitting situation is shown in Figure 2. If a non-trivial factorization of $p_i(x)$ is encountered, say $p_i(x) = p_{i.1}(x)p_{i.2}(x)$, all the polynomials $p_{i'}(x)$, for $i < i' \leq n$, get reduced w.r.t. $p_{i.1}(x)$ and $p_{i.2}(x)$ yielding

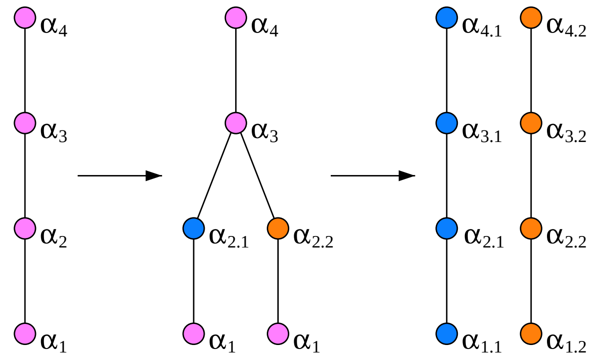


FIGURE 2. Case Splitting during Passive Factorization

new minimal polynomials $p_{i'.1}(x)$ and $p_{i'.2}(x)$. Note, that during this reduction no new case splitting can occur, since we assumed that the $p_{i'}(x)$, $i < i' \leq n$, are already monic and reduced w.r.t. $p_i(x)$. The polynomials $p_{i'}(x)$, for $1 \leq i' < i$, are not affected by this splitting and therefore we have $p_{i'.1}(x) = p_{i'.2}(x) = p_{i'}(x)$, for $1 \leq i' < i$.

## 4. Complexity Analysis: Resolution of one Singularity

In the following we will analyze the computational complexity of the Newton–Puiseux for resolving one curve singularity. The complexity bound will be given in terms of the $y$-degree of the defining polynomial and the Milnor-number of the singularity.

After that, in the next section, we will show how this bound generalizes to a complexity bound for the resolution of all singularities.

Whenever we speak about *the* Puiseux-series of a curve-branch we mean the Puiseux-series computed by the algorithm of Section 2.

We will split the complexity analysis for resolving one singularity into two parts:

First, we will derive an upper bound for the number of operations w.r.t. the algebraically closed field $K$, where we additionally assume that computing all roots of a univariate polynomial is a basic operation in $K$.

Second, the complexity bound w.r.t. $K$ will then be used to compute a complexity bound w.r.t. $k$. Algebraic numbers are handled using passive factorization as described in Section 3. We will neglect additions and subtractions, whereas equality tests play an essential role (splitting of computation) and have to be taken into account.

We assume that the coordinate system has been chosen in such a way that the singular point $q \in \mathbf{A}^2_K$ we want to resolve has coordinates $(0,0)$.

4.1. **Number of Operations in** $K$. Let $\mathfrak{z}_i$, for $1 \leq i \leq I$, be the curve-branches of $\mathcal{C}$ with origin $q$ and lat $\mu$ be the Milnor-number of $q$ w.r.t. $\mathcal{C}$. The sequence of proper neighboring points and the proper multiplicity sequence of $\mathfrak{z}_i$ are denoted by $(q_{i,h})_{0\leq h\leq H_i}$ and $(m_{i,h})_{0\leq h\leq H_i}$, for $1 \leq i \leq I$, respectively. For each curve-branch $\mathfrak{z}_i$ we also define the index-set of free- and satellite-neighboring points and denote them by $H_i^F$ and $H_i^S$.

In the following we use the notation from Definition 4.

A key-point in the complexity analysis below is a bound for the $x$-degrees of the polynomials $f_{i,k}(x,y)$ necessary to compute the singular part of the curve-branches.

**Lemma 4.** *The singular part of the Puiseux-series of $\mathfrak{z}_i$, for $1 \leq i \leq I$, remains unchanged if the polynomials $f_{i,k}(x,y)$, for $0 \leq k \leq K_i$, are truncated at $x$-degrees $M_{i,k}$, where*

$$\begin{aligned} M_{i,k} = l_{i,k} &+ \frac{l_{i,k+1}}{q_{i,k}} + \frac{l_{i,k+2}}{q_{i,k}q_{i,k+1}} \\ &+ \frac{l_{i,k+3}}{q_{i,k}q_{i,k+1}q_{i,k+2}} + \cdots + \frac{l_{i,K_i}}{q_{i,k}\cdots q_{i,K_i-1}}. \end{aligned}$$

*Proof.* The singular part of the Puiseux-series of $\mathfrak{z}_i$ does not change if the polynomial $f_{i,K_i}(x,y)$ is truncated at $x$-degree $M_{i,K_i} = l_{i,K_i}$. Since terms of higher $x$-degree do not contribute to terms of lower $x$-degree when a NT is applied, it follows in general that $f_{i,k}(x,y)$ can be truncated at $x$-degree $l_{i,k} + M_{i,k+1}/q_{i,k}$ without changing the singular part of the Puiseux-series. So we get

$$M_{i,k} = l_{i,k} + \frac{M_{i,k+1}}{q_{i,k}}.$$

Unfolding the recurrence concludes the proof. □

The next Lemma shows that the numbers $M_{i,k}$ are less than or equal to certain sums of multiplicities in the multiplicity-graph of $q$. This enables us to give an upper bound for the total number of coefficients occurring in the singular part computation in terms of $\mu$ and $N$.

First, we have to define certain index-subsets of the neighboring-points of $q$. For this, we recall that a NT can be viewed as a concatenation of one or more QTs. Hencfe, whenever we speak of a neighboring point belonging to a NT, we mean a neighboring point of a QT which is part of a NT.

**Definition 6.** For a curve-branch $\mathfrak{z}_i$, $1 \leq i \leq I$, and a NT $\tau_{i,k}$, $0 \leq k \leq K_i$, we define the *index-set of* $\tau_{i,k}$*-free neighboring points*, written as $H^F_{i,k}$, as

$$H^F_{i,k} = \{h \mid h \in H^F_i \text{ and } q_{i,h} \text{ belongs to } \tau_{i,k}\}.$$

Let $i^*$ be such that $1 \leq i^* \leq I$. The *index-set of* $\tau_{i,k}$*-free neighboring points restricted to the curve-branch* $\mathfrak{z}_{i^*}$, written as $H^F_{i,k;i^*}$, is the set defined as

$$H^F_{i,k;i^*} = \{h \mid h \in H^F_{i,k} \text{ and } q_{i,h} = q_{i^*,h}\}.$$

**Example 2.** We investigate the singularity consisting of the five curve-branches gives by

$$z_1\colon\ x(t) = t^6, \quad y(t) = t^{15} + t^{23},$$

$$z_2\colon\ x(t) = t^6, \quad y(t) = t^{15} - t^{23},$$

$$z_3\colon\ x(t) = t^6, \quad y(t) = t^{15} + t^{19},$$

$$z_4\colon\ x(t) = t^4, \quad y(t) = t^4 + t^7,$$

$$z_5\colon\ x(t) = t^4, \quad y(t) = t^4 + t^9.$$

Its multiplicity-graph is shown in Figure 3.

The following table shows the index-sets of $\tau_{i,k}$-free neighboring points.

| $H^F_{i,k}$ | $z_1$ | $z_2$ | $z_3$ | $z_4$ | $z_5$ |
|---|---|---|---|---|---|
| 0 | $\{0,1,2\}$ | $\{0,1,2\}$ | $\{0,1,2\}$ | $\{0\}$ | $\{0\}$ |
| 1 | $\{4,5,6\}$ | $\{4,5,6\}$ | $\{4,5\}$ | $\{1\}$ | $\{1,2\}$ |

The index-sets of $\tau_{i,k}$-free neighboring points restricted to $z_3$ are given by the table below.

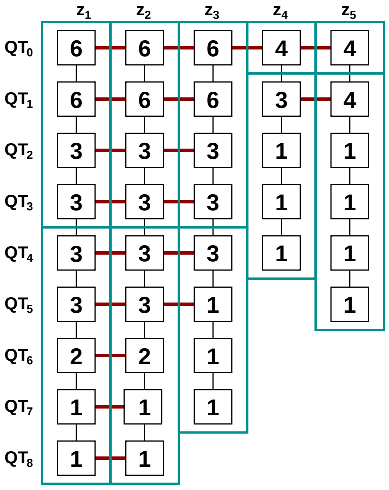


FIGURE 3. Multiplicity Graph

| $H^F_{i,k;3}$ | $z_1$ | $z_2$ | $z_3$ | $z_4$ | $z_5$ |
|---|---|---|---|---|---|
| 0 | $\{0,1,2\}$ | $\{0,1,2\}$ | $\{0,1,2\}$ | $\{0\}$ | $\{0\}$ |
| 1 | $\{4,5\}$ | $\{4,5\}$ | $\{4,5\}$ | $\{\}$ | $\{\}$ |

**Lemma 5.** *For each curve-branch $\mathfrak{z}_i$, $1 \leq i \leq I$, and each step $k$, $0 \leq k \leq K_i$, in the Newton–Puiseux algorithm it holds that*

$$l_{i,k} \leq \sum_{i'=1}^{I} \sum_{h \in H^F_{i',k;i}} m_{i',h}.$$

*Proof.* We fix one curve-branch $\mathfrak{z}_{i_0}$ and one step $k_0$. To simplify notation, we define (see Definition 4): $p_i = p_{i,k_0}$, $q_i = q_{i,k_0}$, $l_i = l_{i,k_0}$, $a_i = a_{i,k_0}$, $b_i = b_{i,k_0}$, and $H_i = H^F_{i,k_0;i_0}$, for $1 \leq i \leq I$. It then holds that

$$l_{i_0} = a_{i_0} + p_{i_0} + \frac{p_{i_0}}{q_{i_0}} b_{i_0},$$

and we have to show that

$$l_{i_0} \leq \sum_{i=1}^{I} \sum_{h \in H_i} m_{i,h}.$$

Furthermore, let $\mu_i = \left\lfloor \frac{p_i}{q_i} \right\rfloor$ and $r_i = p_i - \mu_i q_i$. We assume, that the curve-branches are labeled in such a way that the sequence $\left(\frac{p_i}{q_i}\right)_{1 \leq i \leq I}$ of rational numbers is increasing. This is equivalent to saying that the curve-branches are labeled according to the Newton-polygon.

Note, that we cannot require that the sequence $\left(\frac{p_i}{q_i}\right)_{1 \leq i \leq I}$ is strictly increasing since it may be the case, that two or more curve-branches still correspond to the same segment after $k_0$ steps in the algorithm.

In any case we have

$$\sum_{i=1}^{i_0-1} p_i \geq a_{i_0},$$
$$\sum_{i=i_0+1}^{J} q_i \geq b_{i_0}.$$

As shown in detail in [Stadelmeyer, 2000] the multiplicities of the sequences of free neighboring points belonging to the NTs $\tau_{i,k_0}$ are of the form

$$\begin{aligned} [q_i]^{\mu_i} \quad &, \text{ if } r_i = 0, \\ [q_i]^{\mu_i}[r_i] \quad &, \text{ if } r_i \neq 0. \end{aligned}$$

Because the $\mu_i$ form an ascending sequence, there exists an index $I^*$ with $i_0 \leq I^* \leq I$ such that

$$\begin{aligned} \mu_i \leq \mu_{i_0} \quad &, \text{ for } 1 \leq i \leq I^*, \\ \mu_{i_0} < \mu_i \quad &, \text{ for } I^* < i \leq I. \end{aligned}$$

The multiplicities of the sequences of the neighboring points $q_{i,h}$, for $1 \leq i \leq I$, $h \in H_i$, are then of the form

$$\begin{aligned} [q_i]^{\mu_i} \quad &, \text{ for } 1 \leq i \leq I^* \text{ and } r_i = 0, \\ [q_i]^{\mu_i}[r_i] \quad &, \text{ for } 1 \leq i \leq I^* \text{ and } r_i \neq 0, \\ [q_i]^{\mu_{i_0}}[q_i] \quad &, \text{ for } I^* < i \leq I. \end{aligned}$$

Summing over all curve-branches yields

$$\begin{aligned} \sum_{i=1}^{I} \sum_{h \in H_i} m_{i,h} &= \sum_{i=1}^{I^*} (\mu_i q_i + r_i) + \sum_{i=I^*+1}^{I} (\mu_{i_0} q_i + q_i) \\ &= \sum_{i=1}^{I^*} p_i + (\mu_{i_0} + 1) \sum_{i=I^*+1}^{I} q_i \\ &= \sum_{i=1}^{i_0-1} p_i + p_{i_0} + \sum_{i=i_0+1}^{I^*} \frac{p_i}{q_i} q_i + (\mu_{i_0} + 1) \sum_{i=I^*+1}^{I} q_i. \end{aligned}$$

Using the inequalities $\frac{p_i}{q_i} \geq \frac{p_{i_0}}{q_{i_0}}$, for $i_0 < i \leq I^*$, and $\mu_{i_0} + 1 \geq \frac{p_{i_0}}{q_{i_0}}$ we get

$$\begin{aligned} \sum_{i=1}^{i_0-1} p_i + p_{i_0} + \sum_{i=i_0+1}^{I^*} \frac{p_i}{q_i} q_i + (\mu_{i_0} + 1) \sum_{i=I^*+1}^{I} q_i \geq \\ \sum_{i=1}^{i_0-1} p_i + p_{i_0} + \frac{p_{i_0}}{q_{i_0}} \sum_{i=i_0+1}^{I} q_i \geq \\ a_{i_0} + p_{i_0} + \frac{p_{i_0}}{q_{i_0}} b_{i_0} = l_{i_0}. \end{aligned}$$

□

**Corollary 1.** *For all* $1 \leq i \leq I$ *and* $0 \leq k \leq K_i$ *it holds that*

$$M_{i,k} \leq \sum_{k'=k}^{K_i} \sum_{i'=1}^{I} \sum_{h \in H^F_{i',k';i}} m_{i',h}$$

*and*

$$M_{i,k} \leq \mu + 1,$$

*where* $\mu$ *is the Milnor-number of* $q$.

*Proof.* We fix one curve-branch $\mathfrak{z}_{i_0}$ and one step $k_0$. Since all $q_{i_0,k} \geq 1$, for $0 \leq k \leq K_{i_0}$, we have

$$M_{i_0,k_0} \leq \sum_{k=k_0}^{K_{i_0}} l_{i_0,k} = \sum_{k=k_0}^{K_{i_0}} \sum_{i=1}^{I} \sum_{h \in H^F_{i,k;i_0}} m_{i,h}.$$

To prove the second inequality we recall that (using the notation of the proof of Lemma 5)

$$\sum_{k=k_0}^{K_{i_0}} \sum_{i=1}^{I} \sum_{h \in H^F_{i,k;i_0}} m_{i,h} \leq \sum_{k=k_0}^{K_{i_0}} \sum_{i=1}^{I} (\mu_{i,k} q_{i,k} + r_{i,k}),$$

Since

$$\mu \geq \sum_{k=k_0}^{K_{i_0}} \sum_{i=1}^{I} (\mu_{i,k} q_{i,k}(q_{i,k} - 1) + r_{i,k}(r_{i,k} - 1))$$

it follows from

$$\mu_{i,k} q_{i,k}(q_{i,k} - 1) + r_{i,k}(r_{i,k} - 1) \geq \mu_{i,k} q_{i,k} + r_{i,k},$$

for $q_{i,k} \geq 2$, $r_{i,k} \geq 2$, and

$$\mu_{i,k} q_{i,k}(q_{i,k} - 1) \geq \mu_{i,k} q_{i,k},$$

for $q_{i,k} \geq 2$, that

$$\mu \geq M_{i,k} - 1.$$

□

Disregarding any computational complexity due to algebraic extensions of the ground field as we do now, there is basically one subroutine in the algorithm which contributes to the complexity at each step, namely the procedure `new-polynomial` (see [Duval, 1989]) for a detailed description of the algorithm).

In `new-polynomial` there are at most $(M_{i,k} + 1)(N + 1)$ coefficient multipliciations (and additions) to do. Therefore we get the following bound for the number of $K$-operations necessary to resolve one curve-branch.

**Lemma 6.** *To compute the singular part of the Puiseux-expansion of one curve-branch* $\mathfrak{z}_i$, $1 \le i \le I$, *at most* $\mu(\mu+2)(N+1)$ *operations in* $K$ *are needed. (Not considering any root computation of univariate polynomials.)*

*Proof.* The number $C$ of operations w.r.t. $K$ is bounded by

$$C \le \sum_{k=1}^{K_i} (M_{i,k}+1)(N+1).$$

From Lemma 3 and Corollary 1 we know that $K_i \le \mu$ and $M_{i,k} \le \mu+1$, so we get

$$C \le \mu(\mu+2)(N+1).$$

□

Taking into account that the sum of the multiplicities with which the $x$-degree bounds can be bounded, is non-overlapping for different curve-branches, the complexity bound for resolving one curve-branch at $q$ immediately extends to a complexity bound for resolving all curve-branches at $q$.

**Lemma 7.** *At most* $2\mu(\mu+2)(N+1)$ *operations in* $K$ *are needed to compute the singular parts of all Puiseux-expansion with origin* $q$. *(Not considering any root computation of univariate polynomials.)*

*Proof.* The lemma follows from the fact that the index-sets $H^F_{i,k;i'}$ are essentially non-overlapping in the following sense: If two such index-sets have an index in common, then also the computation in the Newton–Puiseux algorithm is the same for the corresponding neighboring-points (i.e. the curve-branches have not yet been separated and are hence treated together).

But this means, that each free neighboring point of $q$ contributes to the sum of multiplicities only once. Therefore the sum of all $x$-degrees of the transformed polynomials is less than or equal to $\sum_{i=1}^{I} \sum_{h \in H_i^F} m_{i,h}$.

Using the same argumentation as in the proof of Corollary 1 one can show, that this sum is bounded by $2\mu$. □

The next Corollary is not essential for the complexity analysis, but shows how the sum of multiplicities over all free neighboring points is related to the sum over all neighboring points.

**Corollary 2.** *For each curve-branch* $\mathfrak{z}_i$, $1 \le i \le I$, *it holds that*

$$\sum_{h \in H_i^F} m_{i,h} = \sum_{h=1}^{H_i} m_{i,h} + 1.$$

*Proof.* Follows from $m_{i,0} = \sum_{h \in H_i^s} m_{i,h} + 1$ (see Theorem 3 and Lemma 4 in [Stadelmeyer, 2000]). $\square$

A direct consequence of Lemma 5 is the following Corollary from deformation theory.

**Corollary 3.** *The singular part of the Puiseux-series of* $\mathfrak{z}_i$*, for* $1 \le i \le i$*, remains unchanged if the polynomial* $f(x, y)$ *is truncated at total-degree* $\mu + 1$.

*Proof.* As can easily be seen, terms of total-degree greater than $\mu + 1$ in the original polynomial do not contribute to terms of $x$-degree less than $\mu + 1$ in the transformed polynomials. Hence, the result follows from Corollary 1. $\square$

Using the result above we can reformulate Lemma 7 as: The number of K-operations can be bounded by $\mathbf{O}(\mu^3)$. But since $\mu$ may be of order $\mathbf{O}(N^2)$ and the method of deferred expansion relays on the fact that *all* terms up to some given $x$-degree are known, we do not use this result.

4.2. **Basic Operations in** $k$**.** If we handle algebraic numbers the way described in Subsection 3.2 we also have to take into account equality tests between two numbers. The additional advantage of the passive factorization method is, that we can completely neglect root computation, i.e. factorization, which is the reason we never considered it.

The only place, where equality tests are needed is within the procedure `newton-polygon`. Within this procedure we have to test if coefficients become zero or not. Hence at the k-th step there are at most $(M_{i,k}+1)(N+1)$ tests and therefore the total number of equality tests can again be bounded by $\mu(\mu + 2)(N + 1)$ (cf. Lemma 6).

Assuming that the degree of the algebraic extension necessary to compute the Puiseux-series of $\mathfrak{z}_i$ is $e_i$ we know that each operation in K can be replaced by $e_i^2$ operations in $k$. As shown in [Duval, 1989] $\sum_{i=1}^{g} e_i \le N$. Thus a bound for the total number of operations in $k$ would be $\mu(\mu + 2)(N + 1)N^2$. As we will show below, this bound can be improved by a factor of $N^2$, which means, that the additional complexity introduced by the algebraic extensions does not contribute to total complexity of resolving one singularity.

**Lemma 8.** *The number* $K_i$*, for* $1 \le i \le I$*, of NTs for computing the singular part of* $\mathfrak{z}_i$ *is less than or equal to* $\frac{\mu}{e_i}$.

*Proof.* Let $\mathfrak{z}$ be one of the curve-branches $\mathfrak{z}_i$, $1 \le i \le I$, and let $e$ be the degree of the algebraic extension necessary to compute its Puiseux-series. Since we compute $e$ conjugated curve-branches at once, with each NT at least $e$ neighboring points of multiplicity greater or equal to 2 are computed.

The number of these points is at most $\mu$ and therefore the number of steps in the Newton–Puiseux algorithm is at most $\frac{\mu}{e}$. □

**Theorem 1.** *At most* $3\mu^2(N+1)$ *operations in* $k$ *are needed to compute the singular parts of all Puiseux-expansion with origin* $q$.

*Proof.* The proof is basically the same as for Lemma 7. Using the fact, that the sum of multiplicities of conjugated curve-branches bounding the sum of $x$-degrees of the transformed polynomials is non-overlapping, the numbers $M_{i,k}$ have to be less than or equal to $\frac{\mu}{e_i}$. Together with Lemma 8 we get for the total number $C_i$ of operations w.r.t. $k$ necessary to resolve one curve-branch $\mathfrak{z}_i$ the inequality

$$C_i \leq \frac{\mu}{e_i}\left(\frac{\mu}{e_i}+2\right)(N+1)\,e_i^2 = \mu(\mu+2e_i)(N+1).$$

This yields a bound for the total number $C$ of operations w.r.t. $k$ necessary to resolve all curve-branches with origin $q$, which is given by

$$C \leq 3\mu^2(N+1).$$

□

*Remark* 3. It should be possible to improve the above complexity bounds for conjugated curve-branches by a factor of $e_i$. But this is of no practical use.

*Remark* 4. The multiplicities used by the above bound are defined by $p_{i,k}$, $q_{i,k}$ of the NTs and can therefore also be formulated in terms of characteristic Puiseux exponents. (see Theorem of E.-C.)

*Remark* 5. The multiplicities used are exactly those of the first division in the Euclidean algorithm which is on the other hand exactly $p_{i,k}$. But this means that the bound is equal to the highest exponent of the singular part of the Puiseux-series expansion.

## 5. Complexity Analysis: Resolution of all Singularities

Let now $q$ be an arbitrary singular point of the curve $\mathcal{C}$, whose coordinates are algebraic numbers of degree $d$. Moving $q$ to the origin of a new coordinate-system by a linear transformation introduces coefficients of algebraic degree $d$ into the defining polynomial of $\mathcal{C}$ (w.r.t. this new coordinate system).

If we recall the proofs of Section 4 we see that resolving the singularity at $(0,0)$ requires at most $3\mu^2(N+1)d^2$ operations in $k$.

Of course, all conjugated points of $q$ are also singular points of $\mathcal{C}$ and they have the same resolution. Resolving only one representative of a whole conjugated family and using the fact that the sum of the Milnor-numbers over all singular points is bounded by $(D-1)(D-2)$ we get the following Lemma.

**Lemma 9.** *To resolve one (and hence all) of the* $d$ *conjugated singularities of* $\mathcal{C}$ *(at* $(0,0)$*) no more than* $3(D-1)^2(D-2)^2(N+1)$ *operations in* $k$ *are needed.*

*Proof.* As mentioned above the resolution complexity of $q$ is less than $3\mu^2(N+1)e^2$. Since the sum of the Milnor-numbers of the $q$-conjugated family of singular points is at most $(D-1)(D-2)$ (see e.g. [Abhyankar, 1990]), the Milnor-number of $q$ has to be less than or equal to $\frac{(D-1)(D-2)}{e}$. □

If we assume that we have already computed all singular points of $\mathcal{C}$ (which is a standard problem that can be solved by resultant- and gcd-computation), the previous Lemma extends in a straight forward way to a complexity bound for resolving all singularities of $\mathcal{C}$.

**Theorem 2.** *Assuming that all singular points of* $\mathcal{C}$ *are known, the number* $C$ *of* $k$*-operations to resolve all singularities of* $\mathcal{C}$ *is bounded by*

$$C \leq 3(D-1)^2(D-2)^2(N+1) \leq D^5 + \textit{terms of lower order.}$$

*Proof.* Similar to the proof of Lemma 9. □

For a complexity analysis of resultant- and gcd-computations we refer the reader e.g. to [von zur Gathen and Gerhard, 1999].

## 6. Complexity Analysis: System of Adjoint Curves

As explained in [Stadelmeyer, 2000], once all singular points of the curve $\mathcal{C}$ are resolved by the Newton–Puiseux algorithm, computing $\mathrm{Adj}_d(f)$ requires only two more steps, namely:

- First, we have to substitute the singular parts of all curve-branches into a polynomial $g(x,y)$ of degree $d$ with indetermined coefficients obtaining power-series with coefficients depending on the indeterminates of $g(x, y)$.

- Second, setting the coefficients of the power-series to zero up to some calculated order for each curve-branch and collecting the resulting equations, we get a linear system of equations for the coefficients of $g(x, y)$.

Since these are two well-known operations and we know the number of linear equations to solve (there are at most $(D-1)(D-2)$ equations in at most $\frac{D(D+3)}{3}$ unknowns) the complexity analysis for the adjoint computation is basically solved as soon as the complexity for resolving all curve-singularities is known. For this reason we omit further details here.

## 7. Conclusion

As we have seen, the complexity bound for resolving plane curve singularities given by J. Teitelbaum could be improved significantly, namely from $\mathbf{O}(\mu^6)$ to $\mathbf{O}(\mu^3)$. We think, this was made possible by the following reasons:

- Our algorithm is based on Newton transformations instead of quadratic transformations. This allows us to have a very strict control of the $x$-degrees of the intermediate polynomials. Since the $y$-degrees do not increase at all, we can find a rather low bound for the total number of coefficients of the intermediate polynomials occurring in the algorithm.
- The method of multiple variables, combined with passive factorization, seems to be computationally superior to the primitive element method.
- The additional complexity introduced by algebraic numbers occurring in the Newton–Puiseux algorithm is fully covered by the fact, that conjugated curve-branches and singularities can be treated together.
- Using deferred expansion, it is possible to make practical use of the theoretical degree bounds. It also circumvents the problem of having to know some a-priori complexity measure of the singularity, e.g. its Milnor-number, in order to simplify intermediate results without changing its resolution.

## REFERENCES


[Abhyankar, 1990] Abhyankar, S. S. (1990). *Algebraic Geometry for Scientists and Engineers*. American Mathematical Society.

[Brieskorn and Knörrer, 1986] Brieskorn, E. and Knörrer, H. (1986). *Plane Algebraic Curves*. Birkhäuser.

[Dicrescenzo and Duval, 1984] Dicrescenzo, C. and Duval, D. (1984). Computations on curves. In *Lecture Notes in Comput. Sci.*, volume 174, pages 100–107. Springer.

[Dicrescenzo and Duval, 1985] Dicrescenzo, C. and Duval, D. (1985). Algebraic computations on algebraic numbers. In *Informatique et Calcul*, pages 54–61. Wiley-Masson.

[Duval, 1989] Duval, D. (1989). Rational puiseux expansion. *Compositio Math.*, 70:119–154.

[Duval, 1994] Duval, D. (1994). Algebraic numbers: An example of dynamic evaluation. *J. Symbolic Computation*, 18:429–445.

[Newton, 1660] Newton, I. (1660). *Geometria Analitica.*

[Stadelmeyer, 2000] Stadelmeyer, P. (2000). Computing the system of adjoint plane curves by puiseux expansion, part ii: Algorithm. Technical Report 00-04, RISC – Linz, Johannes Kepler Universität, 4040 Linz, Austria, Europe.

[Teitelbaum, 1990] Teitelbaum, J. (1990). The computational complexity of the resolution of plane curve singularities. *Mathematics of Computation*, pages 797–837.

[van der Waerden, 1991] van der Waerden, B. L. (1991). *Algebra, Volume I.* Springer-Verlag.

[von zur Gathen and Gerhard, 1999] von zur Gathen, J. and Gerhard, J. (1999). *Modern Computer Algebra.* Cambridge University Press.

[Walker, 1978] Walker, R. J. (1978). *Algebraic curves.* Springer.

[Walsh, 1998] Walsh, P. G. (1998). A polynomial complexity bound for computations on curves. *SIAM J. Comput.*, 28(2):704–708.

[Walsh, 1999] Walsh, P. G. (1999). On the complexity of rational puiseux expansions. *Pacific J. of Mathematics*, 188(2):369–387.

[Winkler, 1996] Winkler, F. (1996). *Polynomial Algorithms in Computer Algebra.* Springer.