# A dichotomic Newton-Puiseux algorithm using dynamic evaluation

Adrien Poteaux,
CRIStAL
Université de Lille 1
UMR CNRS 9189
Bâtiment M3
59655 Villeneuve d'Ascq, France
adrien.poteaux@univ-lille1.fr

Martin Weimann
LMNO
Université de Caen
UMR CNRS 6139
BP 5186
14032 Caen Cedex
weimann@unicaen.fr

August 29, 2017

Let $F \in \mathbb{K}[X,Y]$ be a polynomial of total degree $D$ defined over a field $\mathbb{K}$ of characteristic zero or greater than $D$. Assuming $F$ separable with respect to $Y$, we provide an algorithm that computes all rational Puiseux expansions of $F$ above $X=0$ in less than $\mathcal{O}(D\,v)$ operations in $\mathbb{K}$, where $v$ is the valuation of the resultant of $F$ and its partial derivative with respect to $Y$. To this aim, we use a divide and conquer strategy and replace univariate factorization by dynamic evaluation. As a first main corollary, we compute the irreducible factors of $F$ in $\mathbb{K}[[X]][Y]$ up to an arbitrary precision $X^N$ with $\mathcal{O}(D(v+N))$ arithmetic operations. As a second main corollary, we compute the genus of the plane curve defined by $F$ with $\mathcal{O}(D^3)$ arithmetic operations and, if $\mathbb{K}=\mathbb{Q}$, with $\mathcal{O}(hD^3)$ bit operations where $h$ is the logarithmic heigth of $F$.

## 1 Introduction

Let $F \in \mathbb{K}[X,Y]$ be a bivariate polynomial defined over a field $\mathbb{K}$. Assuming that the characteristic $p$ of $\mathbb{K}$ is zero or large enough, it is well known that for any $x_0 \in \overline{\mathbb{K}}$, the roots of $F$ (considered as a univariate polynomial in $Y$) may be expressed as fractional Laurent power series in $(X-x_0)$ with coefficients in $\overline{\mathbb{K}}$. These are the (classical) *Puiseux series*[1] of $F$ above $x_0$. Puiseux series are fundamental objects of the theory of algebraic curves [30, 5] and provide important information. The reader will find many applications of Puiseux series in previous papers of the first author (see for instance [26, 25]).

In [11, 12], D. Duval introduced *rational Puiseux expansions*. They are more convenient to compute and provide more geometrical informations (see Section 2). The first contibution of this paper concerns the computations of *singular parts* of rational Puiseux expansions.

In the sequel, we denote $d_X$ and $d_Y$ the partial degrees of $F$ in respectively $X$ and $Y$, and $D$ its total degree. We let also $v = v_X(R_F)$ stands for the $X$-valuation of the resultant $R_F := \mathrm{Res}_Y(F, F_Y))$ of $F$ and its derivative in $Y$ according to the variable $Y$.

**Theorem 1.** *Let $F \in \mathbb{K}[X][Y]$ a primitive and separable polynomial in $Y$ and assume that $p=0$ or $p > d_Y$. There exists an algorithm[2] that computes singular parts of Puiseux series of $F$ above $x_0=0$ in less than $\mathcal{O}(d_Y\,v)$ arithmetic operations.*

This improves results of [26] (the complexity bound therein is in $\mathcal{O}(\rho\, v\, d_Y)$ where $\rho$ is the number of rational Puiseux expansions - bounded by $d_Y$). From that we will deduce:

[1] terms written *in italics* in this introduction will be defined in Section 2 or 5.1.
[2] note that our main algorithms are Las Vegas ; this is only due to the computation of primitive elements ; see Sections 3.1 and 5.2

**Theorem 2.** *Let $F$ as in Theorem 1. There exists an algorithm that computes the singular part of Puiseux series of $F$ above* all *critical points in less than $\mathcal{O}^{\sim}(d_Y{}^2 d_X) \subset \mathcal{O}^{\sim}(D^3)$ arithmetic operations.*

This result improves the bound in $\mathcal{O}^{\sim}(d_Y{}^2 d_X{}^3) \subset \mathcal{O}^{\sim}(D^5)$ of [23, 24] ; note also that Proposition 12 of [26] suggested a way to get a bound in $\mathcal{O}^{\sim}(d_Y{}^3 d_X) \subset \mathcal{O}^{\sim}(D^4)$.

Using the Riemann-Hurwitz formula, we easily deduce:

**Corollary 1.** *Assuming $p = 0$ or $p > D$, there exists an algorithm that computes the genus of a given geometrically irreducible algebraic plane curve over $\mathbb{K}$ of degree $D$ in less than $\mathcal{O}^{\sim}(D^3)$ arithmetic operations.*

Moreover, using the reduction criterion of [23, 25], we can bound the bit complexity of the genus computation (here $\mathrm{ht}(P)$ stands for the maximum between the logarithm of the denominator of $P$, and the logarithm of the infinite norm of its numerator):

**Corollary 2.** *Let $\mathbb{K} = \mathbb{Q}(\gamma)$ be a number field, $0 < \epsilon \leq 1$ a real number and $F \in \mathbb{K}[X, Y]$. Denote $M_\gamma$ the minimal polynomial of $\gamma$ and $w$ its degree. Then there exists a Monte Carlo algorithm that computes the genus of the curve $F(X, Y) = 0$ with probability of error less than $\epsilon$ and an expected number of word operations in:*

$$\mathcal{O}^{\sim}(d_Y{}^2 d_X w^2 \log^2 \epsilon^{-1}[\mathrm{ht}(M_\gamma) + \mathrm{ht}(F)]).$$

With the same notations than in Corollary 2, we have:

**Corollary 3.** *Assuming that the degree of the square-free part of the resultant $Res_Y(F, F_Y)$ is known, there exists a Las Vegas algorithm that computes the genus of the curve $F(X, Y) = 0$ with an expected number of word operations in:*

$$\mathcal{O}^{\sim}(d_Y{}^2 d_X w^2[\mathrm{ht}(M_\gamma) + \mathrm{ht}(F)]).$$

Finally, as described below, our algorithm will induce a fast analytic factorisation of the polynomial $F$. More precisely, one gets the following result:

**Theorem 3.** *Under hypothesis of Theorem 1, there exists an algorithm that computes the irreducible analytic factors of $F$ in $\mathbb{K}[[X]][Y]$ with precision $N \in \mathbb{N}$ in less than $\mathcal{O}^{\sim}(d_Y(\upsilon + N))$ (thus $\mathcal{O}^{\sim}(d_Y N)$ if $N \geq \upsilon$) arithmetic operations in $\mathbb{K}$ plus the cost of one univariate factorisation of degree at most $d_Y$.*

Theorem 3 has a particular interest with regards to factorization in $\mathbb{K}[X, Y]$ or $\overline{\mathbb{K}}[X, Y]$ : when working along a critical fiber, one can take advantage of some combinatorial constraints imposed by ramification when recombining analytic factors into rational factors (see [31]). In particular, if $F$ is irreducible in $\mathbb{K}[[X]][Y]$, then so it is in $\mathbb{K}[X, Y]$.

**Main ideas.** We now describe the main ingredients of our algorithm to improve the results of [26].

**Idea 1.** We concentrate on the monic case. The roots above $(0, \infty)$ require special care, explaining that the results depend on $\upsilon = \upsilon_X(\mathrm{Res}_Y(F, F_Y))$ and not on $\upsilon_X(\mathrm{Disc}_Y(F)) \leq \upsilon$. However, equality holds when $F$ is monic.

**Idea 2.** The key idea is to use tight truncation bounds. Even if the bound $n = \upsilon$ can be reach for *some* Puiseux series, it is actually always possible to compute around half of them using a bound $n \in \mathcal{O}(\upsilon/d_Y)$ (see Section 3). We will use such an idea to show that we can either proves that $F$ is irreducible (and compute its Puiseux series), either compute a factorisation $F = G\,H \mod X^n$ with $n \in \mathcal{O}(\upsilon/d_Y)$ and $d_Y(H) \leq d_Y/2$ ; and that this can be done in the aimed bound of Theorem 2.

**Idea 3.** The fiber $X = 0$ being critical, the polynomials $G(0, Y)$ and $H(0, Y)$ may not be coprime. Therefore, we cannot use the classical Hensel lemma to lift the factorisation of the previous point ; we will show (see Lemma 10) that it is possible to adapt the Hensel lemma to our case.

**Idea 4.** Nevertheless, such a result might require to start with a factorisation $F = G \cdot H \mod X^k$ with $k \in \Theta(\upsilon)$. We will prove in Section 4 that in our context, we will have $k \in \mathcal{O}(\upsilon/d_Y)$.

**Idea 5.** We then apply this divide and conquer strategy until the moment where we get an analytic factorisation of $F \mod X^v$, together with the singular parts of its Puiseux series.

**Idea 6.** Finally, as suggested in the conclusion of [26], computing univariate factorisations needed in the *Newton-Puiseux algorithm* can be costly, in particular when we consider the computation of singular parts of Puiseux series above *all* critical points. As suggested in [26], we will use dynamic evaluation [10, 9] to avoid this bottleneck, which leads us to work over product of fields. As in [12], this will also enable the results to stand over characteristic 0 fields.

**Organisation of the paper.** We remember the classical definitions related to Puiseux series and the description of the *rational Newton Puiseux algorithm* of [12] in Section 2. The remaining of the paper will be divided into two parts:

1. At first, we will assume that the input polynomial is monic and defined above a field ; we will use univariate factorisation, as in [24, 26]. In this context, we will first present in Section 3 a variant of the rational Newton-Puiseux algorithm that will enable us to use tight truncation bounds ; we will use this algorithm to compute half of the Puiseux series above $x_0 = 0$ in $\mathcal{O}(d_Y\, v)$ arithmetic operations plus the cost of univariate factorisations thanks to Idea 2. Then, we show in Section 4 how, starting from these Puiseux series, compute two factors $G$ (corresponding to the Puiseux series computed so far) and $H$ of $F$ with $d_Y(H) \leq d_Y/2$, and then recursively compute the Puiseux series of $H$, getting a divide and conquer algorithm to compute all singular parts of Puiseux series above $x_0 = 0$ in less than $\mathcal{O}(d_Y\, v)$ operations (plus univariate factorisations once again). This will be possible thanks to Ideas 3, 4 and 5. Finally, we will explain in Section 4.5 how to deal with the non monic case as sketched, detailing how to deal with roots above $(0, \infty)$.
2. The next step is to get rid of univariate factorisations. They are costly when the characteristic is 0 and even above finite fields, it has been mentioned in [26] that the cost does not able to get a bound depending on $v$, which is a problem when considering the computation of Puiseux series above *all* critical points. This is Idea 6, i.e. dynamic evaluation. In this context, we are led to work over non integral rings and will have to pay attention to several points ; for instance, the computation of the Newton polygon is not straightforward: we first need to check if the coefficients of the input polynomials are *regular*, and if not to perform some suitable splittings. We detail this in Section 5, where we prove Theorem 1 using results of [9].

Finally, we deduce in Section 6 how all these results together provides us low complexity bounds to compute the desingularisation of the curve above all its critical points, in the same complexity - up to logarithmic factors - than the best known algorithm to compute bivariate resultants (namely, Theorem 2 and associated corollaries). We prove Theorem 3 in Section 7.

**A brief state of the art.** In [12], D. Duval defines the rational Newton-Puiseux algorithm over a field $\mathbb{K}$ with characteristic 0. The complexity analysis therein provides a number of arithmetic operations in less than $\mathcal{O}({d_Y}^6\, {d_X}^2)$ when $F$ is monic (without using any fast algorithm). Note that this algorithm does not compute any univariate factorisation (it uses the D5-principle instead), and can trivially be generalised when $p > d_Y$.

In [23, 24], an algorithm with complexity $\mathcal{O}(d_Y\, v^2 + d_Y\, v\, \log(p^c))$ is provided over $\mathbb{K} = \mathbb{F}_{p^c}$, with $p > d_Y$. Moreover, from this bound, one gets an algorithm that compute the singular parts of Puiseux series of a polynomial $F \in \mathbb{K}[X, Y]$ above *all* critical points in $\mathcal{O}({d_Y}^3\, {d_X}^2\, \log(p^c))$. In [26], with the same assumption on the base field, an algorithm is given to compute the singular part of Puiseux series over $x_0 = 0$ in less than $\mathcal{O}(\rho\, d_Y\, v + \rho\, d_Y \log(p^c))$ arithmetic operations. These two algorithms relie on univariate factorisation over finite fields. Therefore, they cannot be directly extended to the 0 characteristic case. This also explains why the second result does not able to provide an improved bound for the computation of Puiseux series above *all* critical points.

Finally, note that there are other methods to compute Puiseux series or analytic factorisation, as generalised Hensel constructions [16], or the Montes algorithm [21, 2] (which works over general local fields). Several of these methods and a few others have been commented in previous papers of the first author [25, 26]. To our knowledge, none of these method have been proved to provide a complexity which fits in the bounds obtained in this paper.

**Acknowledgment.** This paper is dedicated to Marc Rybowicz, who passed away in November 2016 [13]. The first ideas of this paper actually came from a collaboration between Marc and the first author in the beginning of 2012, that

led to [26] as a first step towards the divide and conquer algorithm presented here. We also thank Francois Lemaire for many useful discussions on dynamic evaluation.

## 2 Main definitions and classical algorithms

### 2.1 Puiseux series

Denote $\mathbb{K}$ a field and consider $F \in \mathbb{K}[X,Y]$ as in Section 1. Up to a change of variable $X \leftarrow X + x_0$, it is sufficient to give definitions and properties for the case $x_0 = 0$. Under the assumption that $p = 0$ or $p > d_Y$, the well known Puiseux theorem asserts that the $d_Y$ roots of $F$ (viewed as a univariate polynomial in $Y$) lie in the field of Puiseux series $\cup_{e\in\mathbb{N}}\overline{\mathbb{K}}((X^{1/e}))$. See [5, 14, 30] or most textbooks about algebraic functions for the 0 characteristic case. When $p > d_Y$, see [8, Chap. IV, Sec. 6]. It happens that these Puiseux series can be grouped according to the field extension they define. Following Duval [12, Theorem 2], we consider decompositions into irreducible elements:

$$\begin{aligned} F &= \prod_{i=1}^{\rho} F_i \text{ with } F_i \text{ irreducible in } \mathbb{K}[[X]][Y] \\ F_i &= \prod_{j=1}^{f_i} F_{ij} \text{ with } F_{ij} \text{ irreducible in } \overline{\mathbb{K}}[[X]][Y] \\ F_{ij} &= \prod_{k=0}^{e_i-1} \left(Y - S_{ij}(X^{1/e_i}\zeta_{e_i}^k)\right) \text{ with } S_{ij} \in \overline{\mathbb{K}}((X)) \end{aligned}$$

where $\zeta_{e_i}$ is a primitive $e_i$-th root of unity in $\overline{\mathbb{K}}$. Primitive roots are chosen so that $\zeta_{ab}^b = \zeta_a$.

**Definition 1.** The $d_Y$ fractional Laurent series $S_{ijk}(X) = S_{ij}(X^{1/e_i}\zeta_{e_i}^k) \in \overline{\mathbb{K}}((X^{1/e_i}))$ are called the *classical Puiseux series* of $F$ above 0. $e_i \in \mathbb{N}$ is the *ramification index* of $S_{ij}$. If $S_{ij} \in \overline{L}[[X^{1/e_i}]]$, we say that $S_{ij}$ *is defined at* $X = 0$.

**Proposition 1.** *The* $\{F_{ij}\}_{1\le j\le f_i}$ *have coefficients in a degree* $f_i$ *extension* $\mathbb{K}_i$ *of* $\mathbb{K}$ *and are conjugated by the action of the Galois group of* $\mathbb{K}_i/\mathbb{K}$. *We call* $\mathbb{K}_i$ *the* residue field *of* $R_i$ *and* $f_i$ *its* residual degree. *We have* $\sum_{i=1}^{\rho} e_i\, f_i = d_Y$.

*Proof.* First claim is [12, Section 1]. For the second one, see e.g. [8, Chapter 4, Section 1]. □

This leads to the definition of rational Puiseux expansions (note that it is possible to construct classical Puiseux series from a system of rational Puiseux expansions - see e.g. [26, Section 2]):

**Definition 2.** A system of *rational Puiseux expansions* over $\mathbb{K}$ ($\mathbb{K}$-RPE) of $F$ above 0 is a set $\{R_i\}_{1\le i\le\rho}$ such that:

- $R_i(T) \in \mathbb{K}_i((T))^2$,
- $R_i(T) = (X_i(T), Y_i(T)) = \left(\gamma_i T^{e_i}, \sum_{l=n_i}^{\infty} \beta_{il} T^l\right)$, $\gamma_i \neq 0$, $n_i \in \mathbb{Z}$
- $R_i$ is a parametrisation of $F_i$, i.e. $F_i(X_i(T), Y_i(T)) = 0$,
- the parametrisation is irreducible, i.e. $e_i$ is minimal.

We call $(X_i(0), Y_i(0))$ the *center* of $R_i$. We have $Y_i(0) = \infty$ if $n_i < 0$, which happens only for non monic polynomials.

Throughout this paper, we will truncate the powers of $X$ of polynomials or series. To that purpose, we introduce the following notation: given $\tau \in \mathbb{Q}$ and a Puiseux series $S$ with ramification index $e$, we denote $\lceil S\rceil^\tau = \sum_{k\le N} \alpha_k X^{k/e}$ where $N = \max\{k \in \mathbb{N} \mid \frac{k}{e} \le \tau\}$. We generalize this notation to elements of $\overline{\mathbb{K}}((X^{1/e}))[Y]$ by applying it coefficient-wise. In particular, if $H \in \mathbb{K}[[X]][Y]$ is defined as $H = \sum_i (\sum_{k\ge 0} h_{ik} X^k) Y^i$, then $\lceil H\rceil^\tau = \sum_i (\sum_{k=0}^{\lfloor\tau\rfloor} h_{ik} X^k) Y^i$.

**Definition 3.** The *regularity index* $r$ of a Puiseux series $S$ of $F$ is the least integer $N \ge \min(0, e\, v_X(S))$ such that if $\lceil S\rceil^{\frac{N}{e}} = \lceil S'\rceil^{\frac{N}{e}}$ for some Puiseux series $S'$ of $F$, then $S = S'$. We call $\lceil S\rceil^{\frac{r}{e}}$ the *singular part* of $S$ in $F$.

Roughly speaking, the regularity index is the number of terms necessary to "separate" a Puiseux series from all the others (with a special care when $v_X(S) < 0$). Since regularity indices of all Puiseux series corresponding to the same rational Puiseux expansion are equal, we define:

**Definition 4.** The *singular part* of a rational Puiseux expansion $R_i$ of $F$ is the pair:

$$\left(\gamma_i T^{e_i}, \Gamma(T) = \sum_{k=n_i}^{r_i} \beta_{ik} T^k\right)$$

where $r_i$ is the regularity index of $R_i$, i.e. the one of any Puiseux series associated to $R_i$.

Once such singular part has been computed, the Implicit Function Theorem ensures us that one can compute the series up to an arbitrary precision. This can be done in quasi linear time by using a Newton operator [17, Corollaries 5.1 and 5.2, page 251].

Finally, we formalise the *precision* of Puiseux series and rational Puiseux expansions that we will use in our algorithms. We also define the notion of pseudo-parametrisation:

**Definition 5.** We say that

- $S' \in \overline{\mathbb{K}}((X^{1/e}))$ is a Puiseux series of $F$ known with precision $n$ if there exists a Puiseux series $S$ of $F$ s.t. $\lceil S' \rceil^n = \lceil S \rceil^n$.
- $R' = (\gamma' T^{e'}, \Gamma'(T)) \in \mathbb{K}((T))^2$ is a RPE of $F$ known with precision $n$ if there exists a RPE $R = (\gamma T^e, \Gamma(T))$ of $F$ such that $e'$ divides $e$, $\gamma'$ divides $\gamma$, $\lceil \Gamma'((X/\gamma')^{1/e'}) \rceil^n = \lceil \Gamma((X/\gamma)^{1/e}) \rceil^n$ for some choices of roots of unity.
- $\pi = (\gamma X^e, \Gamma(X) + \alpha X^\tau Y, )$ is a pseudo-parametrisation of $F$ if $\pi(T, 0)$ is a RPE of $F$ know with precision $\frac{\tau}{e}$.

## 2.2 The rational Newton Puiseux algorithm

Our algorithm in Section 3 is a variant of the well known Newton-Puiseux algorithm [30, 5]. We now explain (roughly speaking) the idea of this algorithm following an example, and finally describe the variant of D. Duval [12, section 4], since we will use the improvements therein.

Let's consider the computation of the Puiseux series of the polynomial $F_0(X, Y) = Y^6 + Y^5X + 5\,Y^4X^3 - 2\,Y^4X + 4\,Y^2X^2 + X^5 - 3\,X^4$. From the well known Puiseux theorem (see e.g. [5, Section 8.3]), we know that the first term of any such series $S(X)$ is of the form $\alpha\, X^{\frac{m}{q}}$, $\alpha \in \overline{\mathbb{K}}$ and $(m, q) \in \mathbb{N}^2$.

We have $F_0(X, \alpha\, X^{\frac{m}{q}} + \cdots) = \alpha^6\, X^{\frac{6\,m}{q}} + \alpha^5\, X^{\frac{5\,m}{q}+1} + 5\,\alpha^4\, X^{\frac{4\,m}{q}+3} - 2\,\alpha^4\, X^{\frac{4\,m}{q}+1} + 4\,\alpha^2\, X^{\frac{2\,m}{q}+2} + X^5 - 3\,X^4 + \cdots$. To get $F_0(X, S(X)) = 0$, at least two terms of the previous sum must cancel, i.e. $(m, q)$ must be chosen so that two of the exponents coincide. To that purpose, we use the following definition:

**Definition 6.** Given $F(X, Y) = \sum_{i,j} f_{ij} X^j Y^i$, $\mathrm{Supp}(F) = \{(i, j) \in \mathbb{N}^2 \mid f_{ij} \neq 0\}$ is the *support* of $F$.

The condition on $(m, q)$ can be translated as: two points of $\mathrm{Supp}(F_0)$ belongs to the same line $m\,a + q\,b = l$. In order to increase the $X$-order of the evaluation, there must be no point under this line. On this example, we have two such lines: $a + 2\,b = 6$ and $a + b = 4$. They define the Newton polygon of $F_0$:

**Definition 7.** The *Newton polygon* $\mathcal{N}(F)$ of $F \in \mathbb{K}[X, Y]$ is the lower part of the convex hull of its support.

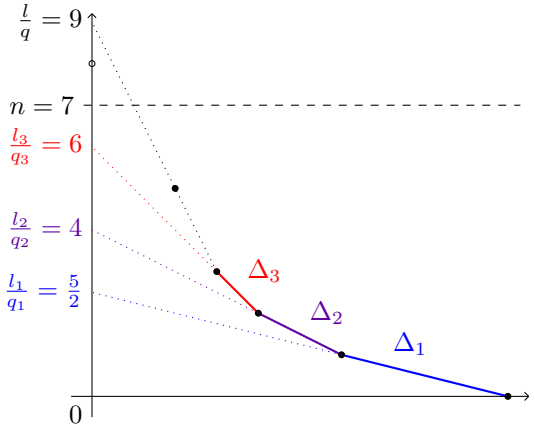
*Remark* 1. In [26, Definition 6], a *modified* Newton polygon $\mathcal{N}^\star(H)$ is defined, and used in the main algorithm. This is not necessary for our strategy, since we do not focus on getting *precisely* the singular part. But we will need that for one proof (see Remark 5), so we use it in the desccription of `RNPuiseux`.

In this paper, we will use low truncation bounds ; in particular, we may truncate some points of the Newton polygon. In order to certify the correctness of the computed slopes, we will use the following definition.

**Definition 8.** Given $F \in \mathbb{K}[X, Y]$ and $n \in \mathbb{N}$, we define the *n-truncated Newton polygon* of $F$ to be the set $\mathcal{N}_n(F)$ of edges $\Delta$ of $\mathcal{N}(\lceil F \rceil^n)$ that satisfies $\frac{l}{q} \leq n$ if $\Delta$ belongs to the line $m\,a + q\,b = l$. In particular, any edge of $\mathcal{N}_n(F)$ is an edge of $\mathcal{N}(F)$.

**Example 1.**

Let us consider $F_1 = Y^{10} + X\,Y^6 + X^2\,Y^4 + X^3\,Y^3 + X^5\,Y^2 + X^8$ and $n = 7$ ; the picture here provides the truncated Newton polygon of $F_1$ with precision 7. Here we have $\lceil F_1 \rceil^n = Y^{10} + X\,Y^6 + X^2\,Y^4 + X^3\,Y^3 + X^5\,Y^2$ and $\mathcal{N}(\lceil F_1 \rceil^n) = [(10,0),(6,1),(4,2),(3,3),(2,5)]$. But the edge $[(3,3),(2,5)]$ is no part of $\mathcal{N}_n(F_1)$, since it belongs to $2\,a + b = 9$, and that there exist points $(i,j)$ so that $2\,i + j \leq 9$ and $j > 7$: from the knowledge of $\lceil F_1 \rceil^7$, we cannot guaranty that $\mathcal{N}(F_1)$ contains an edge belonging to $2\,a + b = 9$. This is indeed wrong here, since $\mathcal{N}(F_1) = [(10,0),(6,1),(4,2),(3,3),(0,8)]$.




We are now considering the choice of $\alpha$ corresponding to $a + 2\,b = 6$. We have $F_0(T^2, \alpha\,T) = (\alpha^6 - 2\alpha^4 + 4\alpha^2)\,T^6 - 3\,T^8 + \alpha^5\,T^7 + (5\alpha^4 + 1)T^{10} + \ldots$, meaning that $\alpha$ must be a non zero root of $P(Z) = Z^6 - 2\,Z^4 + 4Z^2$.

Finally, to get more terms, we can apply recursively this strategy to the polynomial $F_0(X^q, X^m\,(Y + \alpha))$. Obviously, it is more interesting to consider a root $\xi = \alpha^q$ of the polynomial $\phi(Z) = Z^2 - 2\,Z + 4$ (we have $P(Z) = Z^2\,\phi(Z^2)$ and are obviously not interested in a root $\alpha = 0$), that is the characteristic polynomial [12]:

**Definition 9.** If $F = \sum f_{ij} X^j Y^i$, then the *characteristic polynomial* $\phi_\Delta$ of $\Delta \in \mathcal{N}(F)$ is $\phi_\Delta(T) = \sum_{(a,b)\in\Delta} f_{ab} T^{\frac{a-a_0}{q}}$ where $a_0$ is the smallest value such that $(a_0, b_0)$ belongs to $\Delta$ for some $b_0$.

Up to now, we only described the ideas of the algorithm and provided all the necessary tools. We conclude this section by providing a more formal definition of the `RNPuiseux` algorithm for monic polynomials (see Section 4.4 for the non monic case) ; it uses two sub algorithms, for each we only provide specifications:

- `Bézout`, given $(q, m) \in \mathbb{Z}^2$ with $q > 0$, computes $(u, v) \in \mathbb{Z}^2$ s.t. $u\,q - m\,v = 1$ and $0 \leq v < q$.
- `Factor`($\mathbb{K}$,$\phi$) computes the factorisation of $\phi$ over $\mathbb{K}$, given as a list of factors and multiplicities.

**Algorithm:** `RNPuiseux`$(F, \mathbb{K}, \pi)$
**Input:** $F \in \mathbb{K}[X,Y]$ monic, $\mathbb{K}$ a field and $\pi$ the result of previous computations ($\pi = (X,Y)$ for the initial call)
**Output:** A set of singular parts of rational Puiseux expansions above $(0,0)$ of $F$ with its base field.

1 $\mathcal{R} \leftarrow \{\}$; `// results of the algorithm will be grouped in` $\mathcal{R}$
2 **foreach** $\Delta \in \mathcal{N}^\star(F)$ **do** `// we consider only negative slopes`
3  Compute $m, q, l, \phi_\Delta$ associated to $\Delta$;
4  $(u, v) \leftarrow$ `Bézout`$(m, q)$;
5  **foreach** $(\phi, M)$ **in** `Factor`$(\phi_\Delta)$ **do**
6  Take $\xi$ a new symbol satisfying $\phi(\xi) = 0$;
7  $\pi_1 = \pi(\xi^v\,X^q, X^m\,(Y + \xi^u))$;
8  **if** $M = 1$ **then** $\mathcal{R} \leftarrow \mathcal{R} \cup \{(\pi_1, \mathbb{K}(\xi))\}$;
9  **else**
10  $H(X,Y) \leftarrow F(\xi^v X^q, X^m\,(Y + \xi^u))/X^l$ ; `// Puiseux transformation`
11  $\mathcal{R} \leftarrow \mathcal{R} \cup$ `RNPuiseux`$(H, \mathbb{K}(\xi), \pi_1)$;
12 **return** $\mathcal{R}$;

The key improvment of this rational version is the distribution of $\xi$ to both $X$ and $Y$ variables (line 10). This avoids to work with $\alpha = \xi^{1/q}$ and to introduce any useless field extension due to ramification (see [12, Section 4]).

## 2.3 Complexity notions

We finally recall some classical complexity results, starting with the multiplication of univariate polynomials:

**Definition 10.** A (univariate) *multiplication time* is a map $\mathsf{M} : \mathbb{N} \to \mathbb{R}$ such that:

- For any ring $\mathbb{A}$, polynomials of degree less than $d$ in $\mathbb{A}[X]$ can be multiplied in at most $\mathsf{M}(d)$ operations (multiplication or addition) in $\mathbb{A}$
- For any $0 < d \leq d'$, the inequalities $\mathsf{M}(d)\, d' \leq \mathsf{M}(d')\, d$ and $\mathsf{M}(d\, d') \leq \mathsf{M}(d)\, \mathsf{M}(d')$ hold.

**Lemma 1.** *Let $\mathsf{M}$ be a multiplication time. Then we have:*

1. $\mathsf{M}(d + d') \geq \mathsf{M}(d) + \mathsf{M}(d')$ *for any* $d, d' \in \mathbb{N}$,
2. $\mathsf{M}(1) + \mathsf{M}(2) + \cdots + \mathsf{M}(2^{k-1}) + \mathsf{M}(2^k) \leq \mathsf{M}(2^{k+1})$ *for any* $k \in \mathbb{N}$.

*Proof.* The first point is [15, Exercise 8.33]. The second one is an obvious consequence. □

The best known multiplication time gives $\mathsf{M}(d) \in \mathcal{O}(d \log(d) \log(\log(d))) \subset \mathcal{O}^{\sim}(n)$ [28, 6]. Note in particular that we do not have $\mathsf{M}(d)\, \mathsf{M}(d') \leq \mathsf{M}(d\, d')$ but only $\mathsf{M}(d)\, \mathsf{M}(d') \leq \mathsf{M}(d\, d') \log(d\, d')$. This is why we use Kronecker substitution:

**Multiplication of multivariate polynomials.** By Kronecker substitution, one can multiply two polynomials belonging to $\mathbb{A}[Z_1, \cdots, Z_s]$ with degrees in $Z_i$ bounded by $d_i$ in less than $\mathcal{O}(\mathsf{M}(2^{s-1}\, d_1 \cdots d_s))$ operations in $\mathbb{A}$ (it is straightforward to adapt [15, Corollary 8.28, page 247] to any number of variables). In particular, if $s$ is constant, the complexity bound is $\mathcal{O}(\mathsf{M}(d_1 \cdots d_s))$.

**Bivariate polynomials defined over an extension of $\mathbb{K}$.** In particular, in Sections 3 and 4, we perform multiplication of bivariate polynomials in $X$ and $Y$ (with respectives degrees bounded by $d_X$ and $d_Y$) defined over $\mathbb{K}[Z]/(P(Z))$ with $P$ an irreducible polynomial over $\mathbb{K}$ of degree $d_P$ as follows: first perform the polynomial multiplication over $\mathbb{K}[X, Y, Z]$ (in $\mathcal{O}(\mathsf{M}(d_X\, d_Y\, d_P))$ from the previous paragraph) before applying the reduction modulo $P$ on each coefficient (for a total in less than $\mathcal{O}(d_X\, d_Y\, \mathsf{M}(d_P))$ arithmetic operations from [15, Theorem 9.6, page 261]).

Finally, note that we postpone the discussion concerning the complexity of operations modulo triangular sets (needed for dynamic evaluation) in Section 5.2.

# 3 Refined truncation bounds to compute half of the Puiseux series

In all this section, $F \in \mathbb{K}[X, Y]$ *is monic and separable with respect to* $Y$ and the characteristic $p$ of $\mathbb{K}$ satisfies $p = 0$ or $p > d_Y$. Under such hypothesis, we will prove that we can compute at least half of the Puiseux series of $F$ in less than $\mathcal{O}^{\sim}(d_Y\, v)$ arithmetic operations, not counting the factorisation of univariate polynomials:

**Theorem 4.** *There exists an algorithm that computes some RPEs $R_1, \cdots, R_\lambda$ of $F$ known with precision at least $2\, v/d_Y$ and such that $\sum_{i=1}^{\lambda} e_i\, f_i \geq \frac{d_Y}{2}$. Not taking into account univariate factorisations, this can be done in less than $\mathcal{O}(\mathsf{M}(v\, d_Y)\, \log(d_Y)) \subset \mathcal{O}^{\sim}(v\, d_Y)$ arithmetic operations over $\mathbb{K}$.*

Algorithm `Half-RNP` in Section 3.2 will be such an algorithm. It uses previous improvments of the first author and M. Rybowicz [23, 24, 26], and one additional idea, namely Idea 2 of Section 1, that we detail below.

## 3.1 Previous complexity improvments and Idea 2.

Given $P \in \mathbb{K}[Z]$, we denote $\mathbb{K}_P$ the quotient $\mathbb{K}[Z]/(P(Z))$ and $d_P = d_Z(P)$. We need the following results of [24, 26].

**Lemma 2.** *Let $n \in \mathbb{N}$, $F \in \mathbb{K}_P[X, Y]$ and $\xi \in \mathbb{K}_P$ for some irreducible $P \in \mathbb{K}[Z]$. Denote $\Delta$ an edge of $\mathcal{N}(F)$ belonging to $m\, a + q\, b = l$, and $(u, v) = \mathtt{Bézout}(m, q)$. One can reduce the computation of $F(\xi^v X^q, X^m(\xi^u + Y))/X^l$ modulo $X^n$ to $n$ univariate polynomial shifts over $\mathbb{K}_P$. This takes less than $\mathcal{O}(n\, \mathsf{M}(d_Y\, d_P))$ arithmetic operations over $\mathbb{K}$.*

*Proof.* This is [24, Lemma 2, page 210] ; Figure 1 illustrates the idea. Complexity also uses Kronecker substitution. □

In [26], the number of recursive call of the rational Newton-Puiseux algorithm is reduced from $\upsilon$ to $\mathcal{O}(\rho \log(d_Y))$. This is due to what we call the *Abhyankar's trick* [1, Chapter 12]:

**Lemma 3.** *Let $F = Y^{d_Y} + \sum_{i=0}^{d_Y-1} A_i(X)\, Y^i \in \mathbb{K}[X,Y]$. If the Newton polygon of $F(X, Y - A_{d_Y-1}/d_Y)$ has a unique edge $(\Delta)\, m\, a + q\, b = l$ with $q = 1$, then $\phi_\Delta$ has several roots in $\overline{\mathbb{K}}$.*

In other words, after performing the shift $Y \leftarrow Y - A_{d_Y-1}/d_Y$, we can ensure that we will detect at least either a branch separation, a non integer slope, or a non trivial factor of the characteristic polynomial, and this can happen no more than $\mathcal{O}(\rho \log(d_Y))$ times.

**Lemma 4.** *Let $F = Y^{d_Y} + \sum_{i=0}^{d_Y-1} A_i(X)\, Y^i \in \mathbb{K}_P[X,Y]$. One can compute $\lceil F(X, Y - A_{d_Y-1}/d_Y) \rceil^n$ in less than $\mathcal{O}(\mathsf{M}(n\, d_Y(H)\, d_P))$ operations over $\mathbb{K}$.*

*Proof.* From our assumption on the characteristic of $\mathbb{K}$, this computation can be reduced to bivariate polynomial multiplication via [3, Problem 2.6, page 15]. The result follows from Kronecker substitution. □

We only need to apply this trick when the first term of *all* Puiseux series is the same. As in [26], we use in this section the following result, so that we perform such a bivariate shift only if necessary:

**Proposition 2.** *Let $\phi$ be in $\mathbb{K}_P[W]$ with $d = \deg(\phi)$. There exists an algorithm* `Oneroot` *such that* `Oneroot`$(\mathbb{K}_P, \phi)$ *decides if $\phi(T) = c\,(T-\xi)^d$ for some $c, \xi \in \mathbb{K}_P$ using $\mathcal{O}(\mathsf{M}(d\, d_P))$ operations in $\mathbb{K}$.*

*Proof.* Writing $\phi(Y) = \sum_{i=0}^{d} \alpha_i\, Y^i$, it is sufficient to test whether the shifted polynomial $\phi(Y - \alpha_{d-1}/(d\,\alpha_d))$ is equal to $\alpha_d\, Y^d$ ; this can be done in $\mathcal{O}(\mathsf{M}(d\, d_P))$ operations over $\mathbb{K}$ via [3, Problem 2.6, page 15] and Kronecker substitution. □

In order to provide the monicity assumption of Lemma 3, the well-known Weierstrass preparation theorem [1, Chapter 16] is used:

**Proposition 3.** *If $G \in \mathbb{K}_P[X,Y]$ is not divisible by $X$, then there exist unique $\widehat{G}$ and $U$ in $\mathbb{K}_P[[X]][Y]$ such that $G = \widehat{G}\, U$, where $U(0,0) \neq 0$ and $\widehat{G}$ is a Weierstrass polynomial of degree $d_Y(\widehat{G}) = \upsilon_Y(G(0,Y))$. Moreover, RPEs of $G$ and $\widehat{G}$ centered at $(0,0)$ are the same.*

We get a complexity bound for this thanks to the following result:

**Proposition 4.** *Let $G \in \mathbb{K}_P[X,Y]$ be a polynomial satisfying hypotheses of Proposition 3, $n$ be in $\mathbb{N}$ and $\widehat{G}$ denote the Weierstrass polynomial of $G$. There exists an algorithm* `WPT` *such that* `WPT`$(\mathbb{K}_P, G, n)$ *computes $\lceil \widehat{G} \rceil^n$ with $\mathcal{O}(\mathsf{M}(n\, d_Y(G)\, d_P))$ operations in $\mathbb{K}$.*

*Proof.* This is [15, Theorem 15.18, page 451], using Kronecker substitution for multivariate polynomial multiplication. This theorem assumes that $\mathrm{lc}_Y(F)$ is a unit, which is not necessarily the case here. However, formulaes in [15, Algorithm 15.10, pages 445 and 446] can still be applied in our context: this is exactly [19, Algorithm Q, page 33]. □

Finally, in both [24] and [26], we show that powers of $X$ can be truncated modulo $X^{\upsilon+1}$ during the algorithm while preserving the singular part of the Puiseux series. This explains the complexity bound of $\mathcal{O}(\rho\, \upsilon\, d_Y)$ of [26] by taking $n = \upsilon$ in the previous results.

**Representation of residue fields.** As explained in [24, Section 5.1], representing residue fields as multiple extensions induces costly arithmetic cost. Therefore, we need to compute primitive representations each time we get a characteristic polynomial $\phi$ with degree 2 or more. Note that algorithms we use here are Las-Vegas (this is the only probabilistic part concerning our results on Puiseux series computation).

**Proposition 5.** *Let $P \in \mathbb{K}[Z]$ and $\phi \in \mathbb{K}_P[W]$ be two irreducible polynomials of respective degrees $d_P$ and $d_\phi$, and denote $d = d_P\, d_\phi$. Assuming that there are at least $d^2$ elements in $\mathbb{K}$, there exists a Las-Vegas algorithm* `Primitive` *that computes an irreducible polynomial $P_1 \in \mathbb{K}[Z]$ with degree $d$ together with an isomorphism $\Psi : \mathbb{K}_{P,\phi} \simeq \mathbb{K}_{P_1}$ in less than $\mathcal{O}(d^{\frac{\omega+1}{2}})$ arithmetic operations plus a constant number of irreducibility tests in $\mathbb{K}[Z]$ of degree at most $d$. Moreover, given $\alpha \in \mathbb{K}_{P,\phi}$, one can compute $\Psi(\alpha)$ with $\mathcal{O}(d_P\, \mathsf{M}(d))$ operations over $\mathbb{K}$.*

*Proof.* Consider for instance [27, Section 2.2] ; some details are provided in the proof of Proposition 15. □

*Remark* 2. We do not precisely pay attention to the assumption on the number of elements in $\mathbb{K}$ in this paper ; note that we will always have $d \leq d_Y$ in our context, so that if working over a finite field without enough elements, our assumption $p > d_Y$ means that, up to take a degree 2 field extension, we are fine.

**Idea 2: using lower truncations bound.** In this paper, we use tighter truncation bounds: one shows that one can use a truncation bound $n \in \mathcal{O}(\upsilon/d_Y)$ and still get some informations, namely at least half of the singular parts of Puiseux series. But such a truncation bound will need to be updated in a different way than in [24, 26] ; in particular, in case of blow-up $X \leftarrow X^q$, one has to multiply the truncation bound by $q$. In the same way, one cannot divide the truncation bound by $t$ the degree of the found extension anymore. But both these points are actually compensated by the use of the algorithm `WPT`, that will divide the degree in $Y$ by the same amount, since this eliminates all the conjugates. Therefore, the size of the input polynomial will always be bounded by $\mathcal{O}(\upsilon)$ elements of $\mathbb{K}$. Tight truncations bounds (depending on the Puiseux series) are provided in Section 3.3 ; we also provide a bound depending on $\upsilon$.

## 3.2 The `Half-RNP` algorithm.

The following algorithm `Half-RNP` is a variant of algorithm `ARNP` of [26]: the main change is that it we allow to compute only some of the RPEs (at least half of them for our applications); to this aim we update the truncation bound in a slightly different way; also, the computation of the output is presented in a different way (we eliminate the relations of [12, Section 4.1] in another way ; this avoids to deal with field extensions twice), and we include how to deal with primitive elements for sake of completness (see Proposition 5). Correctness of the output is proved in Section 3.3. We denote $v_i := v_X(H_Y(S_{i,j,k}(X)))$ for any Puiseux series $S_{i,j,k}$ associated to $R_i$.

**Algorithm:** `Half-RNP`$(H, P, n, \pi)$

**Input:** $P \in \mathbb{K}[Z]$ irreducible, $H \in \mathbb{K}_P[X, Y]$ separable and monic in $Y$ with $d = d_Y(H) > 0$, $n \in \mathbb{N}$ (truncation order) and $\pi$ the current pseudo-parametrisation ($P = Z$ and $\pi = (X, Y)$ for the initial call).

**Output:** all RPEs of $R_i$ of $H$ such that $n - v_i \geq r_i$, given with precision $(n - v_i)/e_i \geq r_i/e_i$.

1 **if** $H = Y$ **then return** $\pi(T, 0)$;
2 **if** $\mathcal{N}_n(H) = [(0, \alpha), (d, 0)]$ with $\alpha/d \in \mathbb{N}$ **and** `Oneroot`$(\mathbb{K}_P, \phi_\Delta)$ **then** // Abhyankar's trick
3 $\quad B \leftarrow A_{d-1}/d$ ; $\pi_1 \leftarrow \lceil \pi(X, Y - B) \rceil^n$ ; // $H = \sum_{i=0}^{d} A_i(X) Y^i$
4 $\quad$ **if** $d = 1$ **then return** $\pi_1(T, 0)$ **else return** `Half-RNP`$(\lceil H(X, Y - B) \rceil^n, P, n, \pi_1)$;
5 $\mathcal{R} \leftarrow \{\}$; // results will be grouped in $\mathcal{R}$
6 **foreach** $\Delta$ **in** $\mathcal{N}_n(H)$ **do** // $\Delta$ belongs to $m\,a + q\,b = l$ ; $u, v =$ `Bézout`$(m, q)$
7 $\quad$ **foreach** $(\phi, M)$ **in** `Factor`$(\mathbb{K}_P, \phi_\Delta)$ **do** // $\phi_\Delta$ the characteristic polynomial associated to $\Delta$
8 $\quad\quad$ **if** $\deg_W(\phi) = 1$ **then** $\xi, P_1, H_1, \pi_1 = -\phi(Z, 0), P, H, \pi$;
9 $\quad\quad$ **else** // computing a primitive representation
10 $\quad\quad\quad (P_1, \Psi) \leftarrow$ `Primitive`$(P, \phi)$;
11 $\quad\quad\quad \xi, H_1, \pi_1 \leftarrow \Psi(W), \Psi(H), \Psi(\pi)$ ; // $\Psi : \mathbb{K}_{P,\phi} \to \mathbb{K}_{P_1}$ isomorphism
12 $\quad\quad \pi_2 \leftarrow \pi_1(\xi^v X^q, X^m (Y + \xi^u)) \mod P_1$;
13 $\quad\quad H_2 \leftarrow \lceil H_1(\xi^v X^q, X^m (Y + \xi^u)) \rceil^{n_1} \mod P_1$; // $n_1 = q\,n - l$
14 $\quad\quad H_3 \leftarrow$ `WPT`$(H_2, n_1)$;
15 $\quad\quad \mathcal{R} \leftarrow \mathcal{R} \cup$ `Half-RNP`$(H_3, P_1, n_1, \pi_2)$; // no recursive call if $n_1 < 0$
16 **return** $\mathcal{R}$;

*Remark* 3. We have $d_X(\pi) \leq n\,e_i$ for any RPE deduced from $\pi$. This is obvious when $\pi$ is defined from Line 3 ; changing $X$ by $X^q$ on Line 12 is also straightforward. Also, we have $m \leq n\,e_i$ since $\frac{m}{q} \leq \frac{l}{q} \leq n$ from Definition 8.

*Remark* 4. Except possibly at the first call, the polynomial $H$ is then always a Weierstrass polynomial.

Theorem 4 is an immediate consequence of the following result, which will be proved in Section 3.4.

**Theorem 5.** *Let $F \in \mathbb{K}[X,Y]$ be as in Theorem 4. Then* `Half-RNP`$(F, Z, 6\,\upsilon/d_Y, (X,Y)))$ *outputs a set of RPEs, with among them a set $R_1, \cdots, R_\lambda$ known with precision at least $4\,\upsilon/d_Y$ and such that $v_i < 2\,\upsilon/d_Y$ and $\sum_{i=1}^{\lambda} e_i\, f_i \geq \frac{d_Y}{2}$. Not taking into account the cost of univariate factorisations, it takes less than $\mathcal{O}(\mathsf{M}(\upsilon\, d_Y)\, \log(d_Y)) \subset \tilde{\mathcal{O}}(\upsilon\, d_Y)$ arithmetic operations over $\mathbb{K}$.*

### 3.3 Using tight truncations bounds.

We study here more carefully the `RNPuiseux` algorithm in order to get optimal truncation bounds to compute a RPE of a Weierstrass polynomial $F$ with this algorithm or `Half-RNP`. From this study, we also deduce exact relation between $\upsilon$ and these optimal bounds. We will therefore need additional notations:

- $F_Y$ is the derivative in $Y$ of $F$ and $R_1, \cdots, R_\rho$ are its rational Puiseux expansions,
- Let $m_{k,h}\, a + q_{k,h}\, b = l_{k,h}$, $1 \leq h \leq g_k$ be the successive edges encountered during the computation of the expansion $R_k$ with `RNPuiseux`; we denote $N_k = \displaystyle\sum_{h=1}^{g_k} \frac{l_{k,h}}{q_{k,1} \cdots q_{k,h}}$,
- For $1 \leq i \leq \rho$, $r_i, e_i, f_i$ are the data associated to $R_i$, and we let $v_i := v_X(F_Y(S))$ where $S$ is any Puiseux series associated to $R_i$ (this rational number doesn't depends on the choice of $S$).

**Lemma 5.** *With the above notations, for any $1 \leq i \leq \rho$, we have $N_i = \frac{r_i}{e_i} + v_i$.*

*Proof.* Denote $R_i = (\gamma_i\, X^{e_i}, \Gamma_i(X,Y))$ with $\Gamma_i(X,Y) = \Gamma_{i,0}(X) + X^{r_i}\, Y$. By the definition of the Puiseux transformations, the polynomial

$$G_i(X,Y) = \frac{F(\gamma_i\, X^{e_i}, \Gamma_i(X,Y))}{X^{N_i\, e_i}}$$

is such that $(0,1) \in \mathcal{N}(G_i)$, i.e. $v_X\left(\partial_Y G_i(X,0)\right) = 0$. Thus, $v_X(X^{r_i}\, F_Y(\gamma_i\, X^{e_i}, \Gamma_{i,0}(X))) = N_i\, e_i$, or:

$$N_i = \frac{r_i + v_X(F_Y(\gamma_i\, X^{e_i}, \Gamma_{i,0}(X)))}{e_i} = \frac{r_i}{e_i} + v_X(F_Y(X, \Gamma_{i,0}((X/\gamma_i)^{1/e_i}))) = \frac{r_i}{e_i} + v_i. \qquad \square$$

*Remark* 5. From this result, we see that the value $N_i$ actually does not depend on the algorithm. Nevertheless, the proof above rely on the algorithm `RNPuiseux` because it computes *precisely* the singular part of all Puiseux series thanks to the modified Newton polygon of [26, Definition 6]. The algorithm `Half-RNP` introduces two differences:

- the Abhyankar's trick does not change the value of the $N_i$: after applying it, the next value $\frac{l}{q}$ is just the addition of the $\frac{l_i}{q_i}$ we would have found with `RNPuiseux` (the concerned slopes being the sequence of integer slopes that compute common terms for *all* Puiseux series, plus the next one). See Example 2 below.
- not using the modified Newton polygon $\mathcal{N}^\star$ can only change the last value $\frac{l}{q}$, if and only if the coefficient of $X^{r/e}$ is 0. This has no impact on the proof of Lemma 6 below.

In the remaining of this paper, we will define $N_i$ as $\frac{r_i}{e_i} + v_i$.

**Example 2.** Let's assume that $F$ is an irreducible polynomial with Puiseux series $S(X) = X^{1/2} + X + X^{3/2} + X^2 + X^{9/4}$. The successive values for $(l,q)$ are:

- $(4,2)$, $(2,1)$, $(2,1)$, $(2,1)$ and $(2,2)$ with the `RNPuiseux` algorithm. We thus get $N = 2 + 1 + 1 + 1 + \frac{1}{2} = \frac{11}{2}$.
- $(4,2)$, $(14,2)$ with the `Half-RNP` algorithm (assuming high enough truncation). We thus get $N = 2 + \frac{7}{2} = \frac{11}{2}$.

**Lemma 6.** *Let $n_0 \in \mathbb{N}$. To compute the RPE $R_i$ with certified precision $n_0 \geq \frac{r_i}{e_i}$, it is necessary and sufficient to run* `Half-RNP` *with truncation bound $n = n_0 + v_i$. In particular, to ensure the computation of the singular part of $R_i$, it is necessary and sufficient use a truncation bound $n \geq N_i$.*

*Proof.* First note that if we start from a polynomial $H$ known up to $X^n$, then the greatest $n_1$ so that we can certify $H_{\Delta,\xi}$ up to $X^{n_1}$ is precisely $n_1 = q\, n - l$ (see Figure 1 ; details can be found in [24, Proof of Lemma 2, pages 210 and 211]). This explains the truncation update of line 13 in `Half-RNP`.

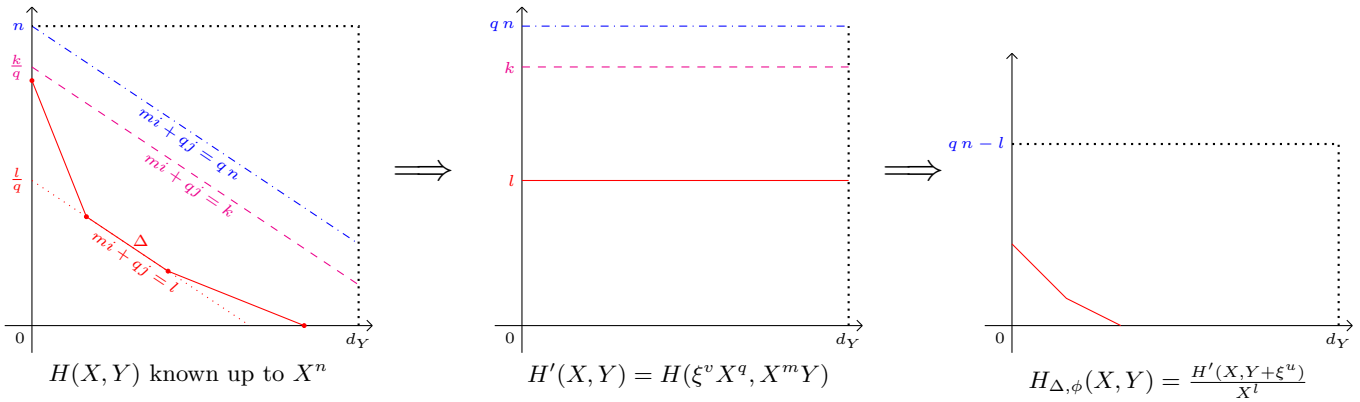



Figure 1: Change of variables for a Puiseux transform

Let's first assume that the coefficient in $X^{\frac{r_i}{e_i}}$ of any Puiseux series asociated to $R_i$ is non zero. Then, starting from a truncation bound $n = n' + N_i$, we get $n_1 = q\,n' + q\,N_i - l$ ; and by definition of $N_i$, $q\,N_i - l$ is precisely the "$N_i$" of the associated RPE of $H_{\Delta,\xi} := H(\xi^v\,X^q, X^m\,(Y + \xi^u))/X^l$. By induction, we finish at the last call of the algorithm associated to the RPE $R_i$ with a truncation bound $n = e_i\,n'$. Moreover, at this point of the algorithm, we have $d_Y(H) = 1$ and $\pi = (\gamma_i\,X^{e_i}, \Gamma_i(X) + \alpha_i\,X^{r_i}\,Y)$. Therefore, the ouptut of the algorithm will be $R_i$ known with precision $n' + \frac{r_i}{e_i}$. Now take $n' = n_0 - \frac{r_i}{e_i}$, and we are done from Lemma 5.

Finally, if the coefficient in $X^{\frac{r_i}{e_i}}$ of any Puiseux series asociated to $R_i$ is zero, we will have $\pi = (\gamma_i\,X^{e_i}, \Gamma_i(X) + \alpha_i\,X^{\eta_i}\,Y)$ with $\eta_i > r_i$. If this is the case, then that means that the at the previous step, we already computed some zero coefficients, thus losing the same precision $\eta_i - r_i$. This does not change the result. □

The quantity $N_i$ is therefore an optimal bound to compute the singular part of the RPE $R_i$. We now bound it.

**Lemma 7.** *We have* $\frac{r_i}{e_i} \leq v_i$.

*Proof.* This is written in the proof of [24, Proposition 5, page 204]. □

**Corollary 4.** *We have* $v_i \leq N_i \leq 2\,v_i$.

*Proof.* Straightforward consequence of Lemmas 5 and 7. □

We finally deduce global bounds:

**Proposition 6.** *There are at least* $\frac{d_Y}{2}$ *Puiseux series* $S_{i,j,k}$ *such that* $v_i < 2\,v/d_Y$ *and* $N_i < 4\,v/d_Y$.

*Proof.* Let's assume that the $R_i$ are ordered by increased $v_i$ values, and define $\lambda$ such that $\sum_{i=1}^{\lambda-1} e_i\,f_i < \frac{d_Y}{2} \leq \sum_{i=1}^{\lambda} e_i\,f_i$ (i.e. $\sum_{i=\lambda+1}^{\rho} e_i\,f_i \leq \frac{d_Y}{2} < \sum_{i=\lambda}^{\rho} e_i\,f_i$ by Proposition 1). Then we have

$$v = \sum_{i=1}^{\rho} v_i\,e_i\,f_i \geq \sum_{i=\lambda}^{\rho} v_i\,e_i\,f_i \geq v_\lambda \sum_{i=\lambda}^{\rho} e_i\,f_i > v_\lambda \frac{d_Y}{2}.$$

The first result is a property of the resultant (see e.g. [15, Exercise 6.12]). The second one is then Corollary 4. □

### 3.4 Complexity results and proof of Theorem 4.

**Proposition 7.** *Not taking into account the cost of univariate factorisations, running* `Half-RNP`$(F, Z, n, (X, Y))$ *with* $F \in \mathbb{K}[X, Y]$ *takes less than* $\mathcal{O}(\mathsf{M}(n\,{d_Y}^2)\log(d_Y))$ *arithmetic operations over* $\mathbb{K}$.

*Proof.* First note that recursive call of line 4 is never done twice in a row thanks to Lemma 3. Representing the set of recursive calls as a tree $\mathcal{T}$, we can group Abhyankar's recursive call with the next Puiseux transform as a single node.

Let's consider a function call to `Half-RNP`$(H, P, n_H, \pi)$ and denote $d_P = \deg_Z(P)$. We distinguish two kind of lines:

(Type 1) By Lemma 4, the number of arithmetic operations for line 4 is bounded by $\mathcal{O}(\mathsf{M}(n\,d_Y))$ using the bound $n\,d_Y \geq n_H\,d_Y(H)\,d_P$. Also, from Remark 3 (and respectively Lemmas 2 and 3), one can bound the cost of Lines 3 and 12 by $\mathcal{O}(\mathsf{M}(n\,d_Y))$.

(Type 2) By respectively Lemma 2 and Proposition 4, lines 13 and 14 are less than $\mathcal{O}(\mathsf{M}(q_\Delta\,d_\phi\,n\,d_Y))$. Thanks to Proposition 5, line 14 fits in the same complexity bound while line 12 costs $\mathcal{O}((d_P d_\phi)^{\frac{\omega+1}{2}})$.

From Lemma 3, when $q = d_\phi = 1$, we must have a branch separation for the corresponding node of $\mathcal{T}$. Therefore, this happens at most $\rho - 1$ times (more precisely, the sum of the nodes of $\mathcal{T}$ of couples $(\Delta, \phi)$ with $q = d_\phi = 1$ is bounded by $\rho - 1$). This means that the sum of the costs for these cases is bounded by $\mathcal{O}(\rho\,\mathsf{M}(n\,d_Y))$.

To conclude the proof, we still have to deal with all the cases where $q > 1$ or $d_\phi > 1$. In such a case, Type 2 lines are the costly ones. Moreover, we can bound $q$ by $e_i$ and $d_P d_\phi$ by $f_i$ for any RPE $R_i$ issued from $(\Delta, \phi)$. But for each RPE $R_i$, such situation cannot happen most than $\log(e_i\,f_i) \leq \log(d_Y)$ times (before and/or after separation of this branch with other ones). From Definition 10, that means we can bound the total cost for all these cases by $\mathcal{O}((\mathsf{M}(\sum_{i=1}^{\rho} e_i\,f_i\,n\,d_Y) + \sum_{i=1}^{\rho} f_i^{\frac{\omega+1}{2}})\log(d_Y)) \subset \mathcal{O}(\mathsf{M}(n\,{d_Y}^2)\log(d_Y))$. □

**Proof of Theorem 5.** As far as correctness is concerned, we only have to take care of truncations and the precision of the output ; other points have been considered in previous papers of the first author [22, 24, 26] (note also [12, Section 4.1] concerning the construction of the output). From Proposition 6 and Lemma 6, we know that we will get at least half of the Puiseux series with precision $4v/d_Y$ or greater and $v_i < 2v/d_Y$ by a function call `Half-RNP`$(F, Z, 6v/d_Y, (X, Y)))$. Finally, the complexity follows from Proposition 7. □

## 4 A dichotomic Newton-Puiseux algorithm.

We now provide a dichotomic strategy to compute all Puiseux series.

**Theorem 6.** *Let $F \in \mathbb{K}[X][Y]$ a primitive and separable polynomial and assume that $p = 0$ or $p > d_Y$. Not taking into acount the cost of univariate factorisations, there exists an algorithm that computes the singular part of all rational Puiseux expansions above $X = 0$ in less than $\mathcal{O}(\mathsf{M}(d_Y\,v)\,\log(d_Y\,v) + \mathsf{M}(d_Y)\,\log^2(d_Y))$ arithmetic operations.*

Assuming that $F$ is monic, our strategy can be summarized as follows:

1. We run `Half-RNP`$(F, Z, 6\,v/d_Y, (X, Y))$. If this provides us all RPEs of $F$, we are done. If not, from Section 3, we get at least half of the Puiseux series of $F$, with precision $4\,v/d_Y$ or more and with $v_i < 2\,v/d_Y$.
2. From these Puiseux series, we construct the associated irreducible factors and their product $G$ with precision $4\,v/d_Y$ ; this is detailed in Section 4.1. Note that $d_Y(G) \geq d_Y/2$ thanks to Theorem 5.
3. We compute its cofactor $H$ by euclidean division modulo $X^{4\,v/d_Y+1}$.
4. We compute a Bezout relation $U\,G + V\,H = X^k \mod X^{k+1}$ using [18, Algorithm 1], proving that there exists such $k$ with $k \leq 2\,v/d_Y$. This is Idea 4, and is detailed in Section 4.2.
5. From this relation and the factorisation $F = G\,H \mod X^{4\,v/d_Y+1}$, we will adapt the Hensel lemma to get a fatorisation $F = G\,H \mod X^{v+1}$ in quasi-linear time. This is Idea 3, detailed in Section 4.3.
6. Finally, we apply our main algorithm recursively on $H$ ; as the degree in $Y$ is at least divided by two each time, this is done at most $\log(d_Y)$ times, for a total cost only multiplied by 2. This is Idea 5, detailed in Section 4.4.

Note that $F$ is not assumed to be monic in Theorem 6. We will need first to use Hensel lifting to compute the factor $F_\infty$ corresponding to RPEs centered at $(0, \infty)$ up to precision $X^v$. Then we will compute the RPEs of $F_\infty$ as "inverse" of the RPEs of its reciprocal polynomial, which is monic by construction. Details are provided in Section 4.5.

### 4.1 Computing the norm of a RPE.

**Lemma 8.** *Let $R_1, \cdots, R_\lambda$ be a set of $\mathbb{K}$-RPEs not centered at $(0,\infty)$. Denote*

$$\nu = \max_{1\le i\le\lambda} \sum_{\substack{(i',j',k')\\ \neq(i,j,k)}} v_X(S_{ijk}(X) - S_{i'j'k'}(X)) \; ; \text{ where for } 1 \le i \le \lambda, \; S_{ijk} \text{ is any Puiseux series associated to } R_i$$

*Assuming that we know the $R_i$ with precision $n \ge \nu$, there exists an algorithm* `NormRPE` *that computes a Weierstrass polynomial $G \in \mathbb{K}[X,Y]$ with $d_Y(G) = \sum_{i=1}^{\lambda} e_i\, f_i$ and $d_X(G) = n + \nu$ such that the RPE of $G$ with precision $n$ are precisely the $R_i$. It takes less than $\mathcal{O}(\mathsf{M}(n\, d_Y(G)^2)\, \log(n\, d_Y(G))) \subset \mathcal{O}^{\sim}(n\, d_Y(G)^2)$ arithmetic operations over $\mathbb{K}$.*

*Proof.* Denote $P_i \in \mathbb{K}[Z]$ so that $R_i = (\gamma_i(Z)\, T^{e_i}, \Gamma_i(Z,T))$ is defined over $\mathbb{K}_{P_i}$. For $1 \le i \le \lambda$, compute

$$A_i = \prod_{j=0}^{e_i-1} \left( Y - \Gamma_i \left( Z, \zeta_{e_i}^j \left( \frac{X}{\gamma_i} \right)^{\frac{1}{e_i}} \right) \right) \mod (X^{n+\nu+1}, P_i(Z))$$

As $n \ge \nu$, this can be done in $\mathcal{O}(\mathsf{M}(e_i^2\, n\, f_i) \log(e_i))$ operations in $\mathbb{K}$ using a subproduct tree. Then, we compute $G_i = \mathrm{Res}_Z(A_i, P_i) \mod X^{n+\nu+1}$ in $\mathcal{O}(f_i\, \mathsf{M}(n\, e_i\, f_i) T \log(n\, e_i\, f_i))$ (just adapt [15, Corollary 11.21, page 332] to a polynomial with three variables). Summing over $i$ these two operations, this fits into our bound. Finally, we compute the product of the $G_i$ modulo $X^{n+\nu+1}$ in less than $\mathcal{O}(\mathsf{M}(n\, d_Y(G)) \log(d_Y(G)))$ using a subproduct tree. □

### 4.2 Lifting order.

Our algorithm requires to lift some analytic factors $G, H$ of $F$ which are not coprime modulo $(X)$. To this aim, we will generalise the classical Hensel lifting. The first step is to compute a generalized Bezout relation $UG + VH = X^\eta$ for some $\eta$ as small as possible.

**Definition 11.** Let $G, H \in \mathbb{K}[[X]][Y]$ coprime. Then the *lifting order* of $G$ and $H$ is defined as:

$$\kappa(G,H) := \inf \{\eta \in \mathbb{N},\ X^\eta \in (G,H)\}.$$

The following proposition gives an upper bound for the lifting order which will be sufficient for our purpose.

**Proposition 8.** *Suppose $H$ monic. We have $\kappa(G,H) \le \max_{H(S)=0} v_X(F_Y(S))$, where $F = G \cdot H$.*

*Proof.* Let assume that $U\,G + V\,H = X^\eta$ in $\mathbb{K}[[X]][Y]$ with $d_Y(U) < d_Y(H) = d$ and denote $S_1, \cdots, S_d$ the Puiseux series of $H$. Then we have $U(S_i)\, G(S_i) = X^\eta$ for $1 \le i \le d$. Using interpolation, we get

$$U = \sum_{i=1}^{d} \frac{X^\eta}{G(S_i)\, H_Y(S_i)} \prod_{j\neq i} (Y - S_j) = \sum_{i=1}^{d} \frac{X^\eta}{F_Y(S_i)} \prod_{j\neq i} (Y - S_j).$$

To have $U \in \mathbb{K}[[X]][Y]$, it suffices that $\eta \ge v_X(F_Y(S_i))$ for $1 \le i \le d$. This concludes ($H$ is monic, so $v_X(S_j) \ge 0$). □

**Corollary 5.** *Assume that $F \in \mathbb{K}[[X]][Y]$ is a non irreducible monic polynomial. Then there exists a factorisation $F = G\,H$ in $\mathbb{K}[[X]][Y]$ such that $\kappa(G,H) \le 2\, v/d_Y$ with $d_Y(G) \ge d_Y/2$.*

*Proof.* From Proposition 6, there exist $\lambda$ RPE $R_1, \cdots, R_\lambda$ of $F$ such that $\sum_{i=1}^{\lambda} e_i\, f_i \ge d_Y/2$ and $v_i < 2\, v/d_Y$. If we consider $G = \prod_{i=1}^{\lambda} F_i$ and $H = \prod_{i=\lambda+1}^{\rho} F_i$, where $F_i$ if the analytic factor associated to $R_i$ (see Section 2.1), then we are done thanks to Proposition 8. □

Finally, note that the relation $U\,G + V\,H = X^\eta \mod X^{\eta+1}$ can be computed in $\mathcal{O}(\mathsf{M}(d_Y\, \eta)\, \log(\eta) + \mathsf{M}(d_Y)\, \eta \log(d_Y))$ from [18, Corollary 1], that is less than $\mathcal{O}(\mathsf{M}(v)\, \log(v))$ if we consider the factorisation of Corollary 5.

### 4.3 Adaptation of Hensel's lemma to our context.

We now generalise the classical Hensel lemma, starting from a relation $F = G\,H$ known modulo $X^{2\,\eta+1}$ with a Bezout relation $U\,G + V\,H = X^\eta \mod X^{\eta+1}$. We refer for instance to [15, section 15.4, page 444] for the classical version when $G$ and $H$ are coprime modulo $(X)$.

**Algorithm:** `HenselStep`$(F, G, H, U, V, n_0, \eta)$

**Input:** $F, G, H, U, V \in \mathbb{K}[Y][X]$, $n_0, \eta \in \mathbb{N}$ s.t. $n_0 > 2\,\eta$, $F = G \cdot H \mod X^{n_0}$, $U \cdot G + V \cdot H = X^\eta \mod X^{n_0-\eta}$, $d_Y(U) < d_Y(H)$ and $d_Y(V) < d_Y(G)$, $H$ monic.

**Output:** $\tilde{G}, \tilde{H}, \tilde{U}, \tilde{V} \in \mathbb{K}[Y][X]$ s.t. $\tilde{G} = G \mod X^{n_0-\eta}$, $\tilde{H} = H \mod X^{n_0-\eta}$, $F = \tilde{G} \cdot \tilde{H} \mod X^{2\,(n_0-\eta)}$, $\tilde{U} \cdot \tilde{G} + \tilde{V} \cdot \tilde{H} = X^\eta \mod X^{2\,n_0-3\,\eta}$, $d_Y(\tilde{V}) < d_Y(\tilde{G})$, $d_Y(\tilde{U}) < d_Y(\tilde{H}) = d_Y(H)$ and $\tilde{H}$ monic.

1 $\alpha \leftarrow X^{-\eta}(F - G \cdot H) \mod X^{2\,(n_0-\eta)}$;
2 $Q, R \leftarrow \mathtt{QuoRem}_Y(U \cdot \alpha, H) \mod X^{2\,(n_0-\eta)}$;
3 $\tilde{G} \leftarrow G + \alpha \cdot V + Q \cdot G \mod X^{2\,(n_0-\eta)}$;
4 $\tilde{H} \leftarrow H + R \mod X^{2\,(n_0-\eta)}$;
5 $\beta \leftarrow X^{-\eta}(U \cdot \tilde{G} + V \cdot \tilde{H}) - 1 \mod X^{2\,n_0-3\,\eta}$;
6 $S, T \leftarrow \mathtt{QuoRem}_Y(U \cdot \beta, \tilde{H}) \mod X^{2\,(n_0-\eta)}$;
7 $\tilde{U} \leftarrow U - T \mod X^{2\,n_0-3\,\eta}$;
8 $\tilde{V} \leftarrow V - \beta \cdot V - S \cdot \tilde{G} \mod X^{2\,n_0-3\,\eta}$;
9 **return** $\tilde{H}$, $\tilde{G}$, $\tilde{U}$, $\tilde{V}$

**Lemma 9.** *Algorithm* `HenselStep` *is correct ; it takes less than* $\mathcal{O}(\mathsf{M}(n_0\,d_Y))$ *operations in* $\mathbb{K}$.

*Proof.* From $\alpha \equiv 0 \mod X^{n_0-\eta}$ (thus $Q \equiv 0 \mod X^{n_0-\eta}$ and $R \equiv 0 \mod X^{n_0-\eta}$ from [15, Lemma 15.9, (ii), page 445]) and $U \cdot G + V \cdot H - X^\eta \equiv 0 \mod X^{n_0-\eta}$, we have $\tilde{G} \equiv G \mod X^{n_0-\eta}$, $\tilde{H} \equiv H \mod X^{n_0-\eta}$ and

$$\begin{aligned} F - \tilde{G} \cdot \tilde{H} &\equiv F - (G + \alpha \cdot V + Q \cdot G) \cdot (H + \alpha \cdot U - Q \cdot H) \\ &\equiv \alpha(X^\eta - V \cdot H - U \cdot G) - \alpha^2 \cdot U \cdot V - Q \cdot \alpha(U \cdot G - V \cdot H) + Q^2 \cdot G \cdot H \equiv 0 \mod X^{2\,(n_0-\eta)}. \end{aligned}$$

From $\beta \equiv 0 \mod X^{n_0-2\,\eta}$ and $U \cdot \tilde{G} + V \cdot \tilde{H} - X^\eta \equiv 0 \mod X^{n_0-\eta}$, we have:

$$\begin{aligned} \tilde{U} \cdot \tilde{G} + \tilde{V} \cdot \tilde{H} - X^\eta &\equiv (U - U \cdot \beta + S \cdot \tilde{H}) \cdot \tilde{G} + (V - \beta \cdot V - S \cdot \tilde{G}) \cdot \tilde{H} - X^\eta \\ &\equiv U \cdot \tilde{G} + V \cdot \tilde{H} - X^\eta - \beta \cdot (U \cdot \tilde{G} + V \cdot \tilde{H}) \\ &\equiv \beta \cdot (X^\eta - U \cdot \tilde{G} - V \cdot \tilde{H}) \equiv 0 \mod X^{2\,n_0-3\,\eta}. \end{aligned}$$

Conditions on the degrees in $Y$ for $\tilde{H}$ and $\tilde{U}$ are obvious (thus is the monicity of $\tilde{H}$). The complexity result is similar to [15, Theorem 9.6, page 261]. □

**Lemma 10.** *Given $F, G, H$ as in the input of algorithm* `HenselStep` *with $n_0 = 2\,\eta + 1$, there exists an algorithm that computes polynomials as in the output of* `HenselStep` *for any precision $n \in \mathbb{N}$, additionally satisfying:*

- $\tilde{G} = G \mod X^{\eta+1}$, $\tilde{H} = H \mod X^{\eta+1}$ *and* $F = \tilde{G} \cdot \tilde{H} \mod X^{n+2\,\eta}$,
- *If* $\exists\ G^\star, H^\star \in \mathbb{K}[Y][X]$ *s.t.* $F = G^\star \cdot H^\star \mod X^{n+2\,\eta}$, *then* $\tilde{G} = G^\star \mod X^n$ *and* $\tilde{H} = H^\star \mod X^n$

*We call* `Hensel` *such an algorithm. It takes less than* $\mathcal{O}(\mathsf{M}(n\,d_Y) + \mathsf{M}(\eta\,d_Y)\,\log(\eta\,d_Y))$ *operations in* $\mathbb{K}$.

*Proof.* First compute $U, V \in \mathbb{K}[X, Y]$ such that $U \cdot G + V \cdot H = X^\eta \mod X^{\eta+1}$ [18, Algorithm 1]. Then, double the value $n_0 - 2\,\eta$ at each call of `HenselStep`, until we get $n_0 - 2\,\eta \geq n + \eta$. Then correctness and complexity are a straightforward consequence of Lemma 9 (using [18, Corollary 1] for the computation of $U$ and $V$). Finally, uniqueness of the result is an adaptation of [15, Theorem 15.14, page 448] (this works because we take a precision such that $n_0 - 2\,\eta \geq n + \eta$). □

*Remark* 6. Note that if $G(0, Y)$ and $H(0, Y)$ are coprime, then $\eta = 0$ and this result is the classical Hensel lemma.

### 4.4 The divide and conquer algorithm for monic polynomials.

We now provide our divide and conquer algorithm. Algorithm `Quo` outputs the quotient of the euclidean division in $\mathbb{K}[[X]][Y]$ modulo a power of $X$.

**Algorithm:** `MonicRNP`$(F, n)$
**Input:** $F \in \mathbb{K}[X, Y]$, separable and monic in $Y$ ; $n \in \mathbb{N}$.
**Output:** If $n$ is big enough, the singular part (at least) of all the RPEs of $F$ above $X = 0$.
1 $\eta \leftarrow \min(n, 6\,n/d_Y)$;
2 $\mathcal{R} \leftarrow$ `Half-RNP`$(F, Z, \eta, (X, Y))$ ;
3 Keep in $\mathcal{R}$ the RPEs with $v_i < \eta/3$; `// These RPEs are known with precision` $\geq 2\eta/3$
4 **if** $\sum_{i=1}^{\lambda} e_i\, f_i = d_Y$ **then return** $\mathcal{R}$; // $\mathcal{R} = \{R_1, \cdots, R_\lambda\}$ ; $e_i$ and $f_i$ associated to $R_i$
5 $G \leftarrow$ `NormRPE`$(\mathcal{R}, 2\eta/3)$;
6 $H \leftarrow$ `Quo`$(F, G, 2\eta/3)$;
7 $G, H \leftarrow$ `Hensel`$(F, G, H, n)$;
8 **return** $\mathcal{R} \cup$ `MonicRNP`$(H, n)$;

**Proposition 9.** *A function call* `MonicRNP`$(F, n)$ *with* $n \geq \upsilon$ *computes all rational Puiseux expansions of* $F$ *above* $0$ *with precision at least* $v_i$ *(in particular, we know the singular parts). Without taking into account the cost of univariate factorisations, it takes less than* $\mathcal{O}(\mathsf{M}(d_Y\, n) \log(d_Y\, n))$ *arithmetic operations.*

*Proof.* From Theorem 5, line 2 takes $\mathcal{O}(\mathsf{M}(n\, d_Y)\, \log(d_Y))$. Moreover, in lines 5 and more, we have $\deg(H) \leq \frac{d_Y}{2}$ and the $R_i$, $1 \leq i \leq \lambda$, are known with precision at least $2\,\upsilon/d_Y$. Then, lines 5, 6 and 7 take respectively less than $\mathcal{O}(\mathsf{M}(n\, d_Y)\, \log(n\, d_Y))$, $\mathcal{O}(\mathsf{M}(n\, d_Y))$ and $\mathcal{O}(\mathsf{M}(n\, d_Y) + \mathsf{M}(\upsilon)\, \log(\upsilon))$ by respectively Lemma 8, division via Newton iteration [15, Theorem 9.4, page 260] and Lemma 10 (we have $F = GH \mod X^{4\,n/d_Y+1}$ by construction and $\kappa(G, H) \leq 2\,\upsilon/d_Y \leq 2\,n/d_Y$ by Corollary 5). This fits in our bound, and the complexity follows by Lemma 1. Precision of the output follows from Proposition 6 and Lemma 6. □

### 4.5 Dealing with the non monic case: proof of Theorem 6.

**Proposition 10.** *There exists an algorithm* `Monic` *that given* $n \in \mathbb{N}$ *and* $F \in \mathbb{K}[X][Y]$ *primitive, returns two polynomials* $F_0, F_\infty \in \mathbb{K}[X][Y]$ *such that* $F = u\, F_0 F_\infty \mod X^n$*, with* $F_0$ *monic in* $Y$*,* $F_\infty(0, Y) = 1$*, and* $u \in \mathbb{K}[X]$*,* $u(0) \neq 0$ *with at most* $\mathcal{O}(\mathsf{M}(nd_Y))$ *operations over* $\mathbb{K}$*.*

*Proof.* This is [19, Algorithm Q, page 33] (see the proof of Proposition 4). □

We can now give our main algorithm `RNP` which will compute the singular part of all RPEs of $F$ above $X = 0$, including those centered at $(0, \infty)$.

**Algorithm:** `RNP`$(F, n)$
**Input:** $F \in \mathbb{K}[X, Y]$, separable in $Y$ and $n \in \mathbb{N}$.
**Output:** If $n$ is big enough, the singular part (at least) of all the RPEs of $F$ above $X = 0$
1 $(F_0, F_\infty) \leftarrow$ `Monic`$(F, n)$;
2 $F'_\infty \leftarrow Y^{d_Y(F_\infty)} F_\infty(X, 1/Y)$;
3 $\mathcal{R}_\infty \leftarrow$ `MonicRNP`$(F'_\infty, n)$;
4 Inverse the second element of each $R \in \mathcal{R}_\infty$;
5 **return** `MonicRNP`$(F_0, n) \cup \mathcal{R}_\infty$;

The proof of Theorem 6 follows immediately from the following proposition :

**Proposition 11.** *Assume that* $p = 0$ *or* $p > d_Y$*. Not taking into acount the cost of univariate factorisations, a function call* `RNP`$(F, \upsilon)$ *returns the expected output with at most* $\mathcal{O}(\mathsf{M}(d_Y\, \upsilon)\, \log(d_Y\, \upsilon))$ *arithmetic operations.*

There is one delicate point in the proof of Proposition 11: we need to invert the RPEs of $F'_\infty$ and it is not clear that the truncation bound $n = v$ is sufficient for recovering in such a way the singular part of the RPEs of $F_\infty$ (see also Remark 7 below). We will need the following two results:

**Lemma 11.** *Given two distinct Puiseux series $S$ and $S'$, we have $v_X\left(\frac{1}{S} - \frac{1}{S'}\right) = v_X(S - S') - v_X(S) - v_X(S')$.*

*Proof.* If $v_X(S) \neq v_X(S')$, one can assume up to renaming the series that $v_X(S) < v_X(S')$. Then $v_X(S - S') = v_X(S)$ and $v_X\left(\frac{1}{S} - \frac{1}{S'}\right) = v_X(S')$. If $v_X(S) = v_X(S') = \alpha$, then $\frac{X^\alpha}{S} \mod X^n$ is uniquely determined from $X^{-\alpha} S \mod X^n$ (same for $S'$). Denoting $s = v_X(S - S') - \alpha$, then we have $X^{-\alpha} S = X^{-\alpha} S' \mod X^s$, thus $\frac{X^\alpha}{S} = \frac{X^\alpha}{S'} \mod X^s$ and $s + v_X\left(\frac{1}{S} - \frac{1}{S'}\right) \geq s$. Similarly, denoting $s_0 = \alpha + v_X\left(\frac{1}{S} - \frac{1}{S'}\right)$, we get $v_X(S - S') - s \geq s_0$, concluding the proof. □

**Proposition 12.** *Let $R_i = (\lambda_i X^{e_i}, \Gamma_i)$ be an RPE of $F_\infty$ and let $s_i = v_X(\Gamma_i)$ (so $s_i < 0$). The function call* `MonicRNP`$(F'_\infty, v_{F_\infty})$ *computes each RPE $R'_i$ with precision at least $\frac{r_i - 2\,s_i}{e_i}$.*

*Proof.* Let $d = d_Y(F_\infty)$ and let $v$ stands for the $X$-valuation of the leading coefficient of $F_\infty$ with respect to $Y$. Let $S_1, \cdots, S_d$ the Puiseux series of $F_\infty$ and $S_i$ one of the Puiseux series associated to the RPE $R_i$ of $F_\infty$ we are considering. Note that $\frac{s_i}{e_i} = v_X(S_i)$. By definition of the regularity index, we have $v_X(S_i - S_j) \leq \frac{r_i}{e_i}$ for all $j \neq i$. Let's denote $i_0$ such that $v_X(S_i - S_{i_0}) = \max\limits_{j \neq i} v_X(S_i - S_j)$ (several values of $i_0$ are possible). We distinguish three cases:

1. $v_X(S_i) = v_X(S_{i_0})$. Then we either have $v_X(S_i - S_{i_0}) = \frac{r_i}{e_i}$, or $e_{i_0} = q\, e_i$ with $q$ an integer $> 1$, and there exist $q$ conjugates Puiseux series $(S_{i_0}^{[j]})_{0 \leq j < q}$ of $S_{i_0}$ such that $v_X(S_i - S_{i_0}) = v_X(S_i - S_{i_0}^{[j]})$, implying $\sum_{j=0}^{q-1} v_X(S_i - S_{i_0}^{[j]}) \geq \frac{r_i}{e_i}$ ; see [24, Case 3 in Proof of Proposition 5, pages 204 and 205] for details.
2. $v_X(S_i) > v_X(S_{i_0})$ ; then $r_i = s_i$ and $v_X(S_i) > v_X(S_j)$ for any $j \neq i$ by definition of $i_0$. In particular, this induces $\frac{v}{d} \geq -\frac{s_i}{e_i}$ since $v = \sum_{i=1}^{d} -\frac{s_i}{e_i}$ from [24, Lemma 1, page 198].
3. $v_X(S_i) < v_X(S_{i_0})$ ; then $v_X(S_i - S_{i_0}) = s_i = r_i$. We can also assume that $v_X(S_j) \neq v_X(S_i)$ for all $j \neq i$: if $v_X(S_j) = v_X(S_i)$, then by definition of $i_0$ we have $v_X(S_i - S_j) = v_X(S_i - S_{i_0})$ and one could have define $i_0 = j$ and deal with it as Case 1.

We can now prove Proposition 12. First, when considering Case 1, knowing each $\frac{1}{S_i}$ with precision $v'_i := v_X\left(\partial_Y F'_\infty\left(\frac{1}{S_i}\right)\right)$ is sufficient: from Lemma 11, we have $v'_i = \sum_{j \neq i} v_X(\frac{1}{S_i} - \frac{1}{S_j}) = \sum_{j \neq i} v_X(S_i - S_j) - v_X(S_i) - v_X(S_j)$. As either $v_X(S_i - S_{i_0}) - v_X(S_i) - v_X(S_{i_0}) = \frac{r_i - 2\,s_i}{e_i}$ or $\sum_{j=0}^{q-1} v_X(S_i - S_{i_0}^{[j]}) - v_X(S_i) - v_X(S_{i_0}^{[j]}) \geq \frac{r_i - 2\,s_i}{e_i}$, we are done.

Then, concerning Case 2, from Proposition 9 and Lemma 6, we know the RPE $R_i$ with precision at least $v_i + \frac{v}{d_Y}$, that is at least $\frac{v}{d_Y}$ thus $-\frac{s_i}{e_i}$ from the discussion above. As $r_i = s_i$, this is at least $\frac{r_i - 2\,s_i}{e_i}$.

Finally, Case 3 requires more attention. Let's first assume that $v_X(S_i) > v_X(S_j)$ for some $j \neq i$ ; then $v_X(S_i - S_j) - v_X(S_i) - v_X(S_j) = v_X(S_i) = -\frac{s_i}{e_i}$, and we are done since $r_i = s_i$. If not, then we have $v_X(S_i) < v_X(S_j)$ for all $j$. This means that $e_i = f_i = 1$, and that $\mathcal{N}(H)$ for $H$ a factor of $F'_\infty$ in $\mathbb{K}[[X]][Y]$ has an edge $[(0, \alpha - s_i), (1, \alpha)]$. We just need to prove that the truncation bound used when dealing with this Puiseux series is at least $\alpha - s_i$. As long as this is not the case, this edge is not considered from the definition of $\mathcal{N}_n(H)$ ; also, at each recursive call of `MonicRNP` (Line 8), the value of $\alpha$ decreases and the truncation bound increased (since $d_Y$ is at least divided by 2). If the truncation bound is not high enough before, we will end with a polynomial of degree 1, corresponding to $\alpha = 0$, and a truncation bound equal to $n = v$, that is more than $-s_i$. This concludes. □

*Proof of Proposition 11.* Let us show that the truncation bound for $F'_\infty$ is sufficient for recovering the singular part of the Puiseux series of $F_\infty$. First note that the inversion of the second element is done as follows: given $R'_i(T) = (\gamma_i\, T^{e_i}, \Gamma'_i(T) = \sum_{k=0}^{\tau_i} \alpha'_{i,k}\, T^k)$ and denoting $s_i = -v_T(\Gamma'_i(T)) < 0$, we compute the inverse of $T^{s_i}\, \Gamma'_i(T)$ (that has a non zero constant coefficient) via quadratic Newton iteration [15, Algorithm 9.3, page 259] in less than $\mathcal{O}(\mathsf{M}(\tau_i + s_i))$ arithmetic operations [15, Theorem 9.4, page 260]. In order to get the singular part of the corresponding RPE $R_i$ of $F_\infty$, we need to know $R_i$ with precision $\frac{r_i}{e_i}$, i.e. to know at least $r_i - s_i + 1$ terms. That means that we need to know $R'_i$ with precision $r_i - 2\,s_i$ or more. This holds thanks to Proposition 12. Correctness and complexity of Algorithm `RNP` then follows straightforwardly from Proposition 9 and Proposition 10. □

*Remark* 7. Note that precision $v_X(\mathrm{Disc}_Y F)$ is not always enough to get the singular part of the Puiseux series centered at $(0, \infty)$, as shows the following example: consider $F(X, Y) = 1 + X\, Y^{d-1} + X^{d+1}\, Y^d$, that has a reciprocal polynomial

$F' = Y^d + XY + X^{d+1}$. Here we have $v_X(\mathrm{Disc}_Y F) = d$, and $\lceil F' \rceil^d = Y^d + XY$. This is enough to compute the singular part of Puiseux series of $F'$, but not to get one term of the Puiseux series $X^d$. Nevertheless, the precision $v = v + v_X(\mathrm{Disc}_Y F) = 2d + 1$ is sufficient.

*Proof of Theorem 6.* The algorithm mentionned in this theorem is `RNP` described above, run with parameter $(F, v)$. We can compute the parameter $v$ with at most $\mathcal{O}(\mathsf{M}(d_Y v)\log(d_Y v)$ operations from [18, Corollary 1]. □

*Remark* 8. Another way to approach the non monic case is the one used in [24]. The idea is to use algorithms `MonicRNP` and `Half-RNP3` even when $F$ is not monic. This would change nothing as far as these algorithms are concerned, but the proof concerning truncation bounds must be adapted: defining $s_i = \min(0, v_X(S_i))$, $N'_i = N_i - \frac{s_i}{e_i} d_Y$, $v'_i = v_i - \frac{s_i}{e_i} d_Y$, we can show that $N'_i = \frac{r_i}{e_i} + v'_i$, using [24, Figure 3, page 211] to deal with the possible positive slopes of the initial call. Then the remaining results of Section 3.3 are preserved, replacing $v_i$ by $v'_i$ and $N_i = N'_i$, using some intermediate results of [24] (in particular, to prove $\frac{r_i}{e_i} \leq v'_i$, we need to use some formulas in the proof of [24, Proposition 5, page 204]). We chose to consider the monic case separately, since it makes one of the main technical results of this paper (namely tight truncation bounds) less difficult to apprehend, thus the paper more progressive to read.

## 5 Avoiding univariate factorisation

As pointed out in the conclusion of [26], even over finite fields (when considering *all* critical points), the cost of univariate factorisation cannot be bounded so that it fits into our objectives. Up to now, we proved Theorem 1 up to the cost of univariate factorisations. To conclude the proof, one would additionally need to prove that the cost of all univariate factorisations computed when calling Algorithm `Half-RNP` is in $\mathcal{O}^{\sim}(v\, d_Y)$. As $v$ can be small, that means that we would need a univariate factorisation algorithm for a polynomial in $\mathbb{K}[Y]$ of degree at most $d$ with complexity $\mathcal{O}^{\sim}(d)$. Unfortunately, this does not exists. We will solve this point via Idea 6: another approach is the one of Della Dora, Dicrescenzo and Duval, who introduced the "dynamic evaluation" techniques [10, 9] (also named "D5 principle") as a mean to compute with algebraic numbers, while avoiding factorisation (replacing it by *squarefree* factorisation). This technique will lead us to consider polynomials with coefficients belonging to a direct product of field extensions of $\mathbb{K}$ ; more precisely to a zero-dimensional *non integral* $\mathbb{K}$-algebra $\mathbb{K}_I = L[Z_1, \cdots, Z_s]/I$ where $I$ is defined as a triangular set in $\mathbb{K}[\underline{Z}] := \mathbb{K}[Z_1, \cdots, Z_s]$. In this context, zero-divisors might appear, causing triangular decomposition and splittings (see Section 5.1 for details). In our context, three main subroutines of the `Half-RNP` algorithm can lead to a decomposition of the coefficient ring:

(i) computation of Newton polygons,

(ii) squarefree factorisations of characteristic polynomials,

(iii) computation of primitive elements.

There are two other points that we need to take care of for our main program:

(iv) subroutine `Hensel`, via the initial use of [18, Algorithm 1],

(v) the initial factorisation of algorithm `RNP`.

To simplify the comprehension of this section, we will not mention logarithm factors in our complexity results, using only the $\mathcal{O}^{\sim}$ notation. This section is divided as follows:

1. We start by recalling a few definitions on triangular sets and in particular our notion of *D5 rational Puiseux expansions* in Section 5.1.
2. The key point of this section is to deal with these splitting with almost linear algorithms ; to do so, we mainly rely on [9]. We briefly review in Section 5.2 their results ; additionally, we introduce a few algorithms needed in our context. In particular, this section details points (iii) and (iv) above.
3. Points (i) and (ii) above are grouped in a unique procedure `Polygon-Data`, detailed in Section 5.3.
4. We provide the D5 versions of algorithms `Half-RNP`, `MonicRNP` and `RNP` in Section 5.4.
5. Finally, we prove Theorem 1 in Section 5.5.

## 5.1 Triangular sets and dynamic evaluation

**Definition 12.** We call a (monic, autoreduced) *triangular set* of $\mathbb{K}[Z_1, \cdots, Z_s]$ a set of polynomials $P_1, \cdots, P_s$ such that:

- $P_i \in \mathbb{K}[Z_1, \cdots, Z_i]$ is monic in $Z_i$,
- $P_i$ is reduced modulo $(P_1, \cdots, P_{i-1})$,
- the ideal $(P_1, \cdots, P_s)$ of $\mathbb{K}[\underline{Z}]$ is radical.

We abusively call an ideal $I \subset \mathbb{K}[\underline{Z}]$ a triangular set if it can be generated by a triangular set $(P_1, \ldots, P_s)$. We denote by $\mathbb{K}_I$ the quotient ring $\mathbb{K}[\underline{Z}]/(I)$.

Note that this defines a zero-dimensional lexicographic Gröbner basis for the order $Z_1 < \cdots < Z_s$ with a triangular structure. Such a product of fields contains zero-divisor:

**Definition 13.** We say that a non-zero element $\alpha \in \mathbb{K}_I$ is *regular* if it is not a zero-divisor. We say that a polynomial or a parametrisation is defined over $\mathbb{K}_I$ is regular if all its non zero coefficients are regular.

Given a zero-divisor $\alpha$ of $\mathbb{K}_I$, one can divide $I$ as $I = I_0 \cap I_1$ with $I_0 + I_1 = (1)$, $\alpha \mod I_0 = 0$ and $\alpha \mod I_1$ is invertible. Moreover, both ideals $I_0$ and $I_1$ can be represented by triangular sets of $\mathbb{K}[\underline{Z}]$.

#### Triangular decomposition.

**Definition 14.** A *triangular decomposition* of an ideal $I$ is $I = I_1 \cap \cdots \cap I_k$ such that $I_i + I_j = (1)$ for any $1 \leq i \neq j \leq k$, and such that every $I_i$ can be represented by a triangular set.

Thanks to the Chinese remainder theorem, the $\mathbb{K}$-algebra $\mathbb{K}_I$ is isomorphic to $\mathbb{K}_{I_1} \oplus \cdots \oplus \mathbb{K}_{I_k}$ for any triangular decomposition of $I$. We extend this isomorphism coefficient wise for any polynomial or series defined above $\mathbb{K}_I$.

**Definition 15.** Considering any polynomial or series defined above $\mathbb{K}_I$, we defined its *splitting* according to a triangular decomposition $I = I_1 \cap \cdots \cap I_k$ the application of the above isomorphism coefficient-wise.

A key point as far complexity is concerned is the concept of *non critical* triangular decompositions. We recall [9, Definitions 1.5 and 1.6]:

**Definition 16.** Two polynomials $a, b \in \mathbb{K}_I[X]$ are said *coprime* if the ideal $(a, b) \subset \mathbb{K}_I[X]$ is equal to (1).

**Definition 17.** Let $(P_1, \cdots, P_s)$ and $(P'_1, \cdots, P'_s)$ be two distinct triangular sets. We define the *level* $l$ of this two triangular sets to be the least integer such that $P_l \neq P'_l$. We say that these triangular sets are *critical* if $P_l$ and $P'_l$ are not coprime in $\mathbb{K}[Z_1, \cdots, Z_{l-1}]/(P_1, \cdots, P_{l-1})$. A triangular decomposition $I = I_1 \cap \cdots \cap I_k$ is said *non critical* if it has no critical pairs ; otherwise, it is said *critical*.

**D5 rational Puiseux expansions.** We conclude this section by defining systems of D5-RPEs over fields and product of fields. Roughly speaking, a system of D5-RPE over a field $\mathbb{K}$ is a system of RPEs over $\mathbb{K}$ grouped together with respect to some square-free factorisation of the characteristic polynomials, hence without being necessarily conjugated over $\mathbb{K}$. We have to take care of two main points. Since we want to get informations before fields splittings, the computed parametrisations by our algorithm will be regular (i.e. not containing any zero divisors). And of course, we want to recover usual system of RPEs after fields splittings. In particular, the computed parametrisation will fit the following definition:

**Definition 18.** Let $F \in \mathbb{K}[X, Y]$ be a separable polynomial over a field $\mathbb{K}$. A system of *D5 rational Puiseux expansions* over $\mathbb{K}$ of $F$ above 0 is a set $\{R_i\}_{1 \leq i \leq \rho}$ such that:

- $R_i \in \mathbb{K}_{P_i}((T))^2$ for some square-free polynomial $P_i$,
- Denoting $P_i = \prod_j P_{ij}$ the univariate factorisation of $P_i$ over $\mathbb{K}$ and $\{R_{ij}\}_j$ the splitting of $R_i$ according to the decomposition $\mathbb{K}_{P_i} = \oplus_j \mathbb{K}_{P_{ij}}$, then the set $\{R_{ij}\}_{i,j}$ is a system of $\mathbb{K}$-RPE of $F$ above 0 (as in Definition 2).

In order to deal with *all* critical points in Section 6, we will not compute the RPE's of $F$ above 0 but above a root of a squarefree factor $Q$ of the resultant $R_F$. This leads us to the following definition:

**Definition 19.** Let $F \in \mathbb{K}_Q[X,Y]$ separable for some $Q \in \mathbb{K}[X]$ squarefree. We say that $F$ admits a system of D5-RPE's over $\mathbb{K}_Q$ above 0 if there exists parametrisations as in Definition 18 that are regular over $\mathbb{K}_Q$.

Now, a system of *D5 rational Puiseux expansions* over $\mathbb{K}$ of $F$ above the roots of $Q$ is a set $\{Q_i, \mathcal{R}_i\}_i$ such that:

- $Q = \prod_i Q_i$,
- $\mathcal{R}_i$ is a system of D5 RPE's over $\mathbb{K}_{Q_i}$ of $F(X+Z,Y) \mod Q_i(Z)$ above 0 (in the sense of definition above).

## 5.2 Complexity of dynamic evaluation.

**Results of [9].** We start by recalling the main results of [9], providing them only with the $\mathcal{O}^{\sim}$ notation (i.e. forgetting logarithm factors). In particular, we will take $\mathsf{M}(d) \in \mathcal{O}^{\sim}(d)$ in the following. In our paper, we also assume the number of variables defining triangular sets to be constant (we usually have $s = 2$ in our context).

**Definition 20.** An *arithmetic time* is a function $I \mapsto \mathsf{A}_s(I)$ with real positive values and defined over all triangular sets in $\mathbb{K}[Z_1, \cdots, Z_s]$ such that the following conditions hold:

1. For every triangular decomposition $I = I_1 \cap \cdots \cap I_h$, we have $\mathsf{A}_s(I_1) + \cdots + \mathsf{A}_s(I_h) \leq \mathsf{A}_s(I)$.
2. Any addition or multiplication in $\mathbb{K}_I$ can be made in $\mathsf{A}_s(I)$ operations over $\mathbb{K}$,
3. Given a triangular decomposition of $I = I_1 \cap \cdots \cap I_h$ in less than $\mathsf{A}_s(I)$ arithmetic operations, one can compute a non-critical triangular decomposition of $I$ that refines it. We denote `removeCriticalPairs` such an algorithm.
4. Given $\alpha \in \mathbb{K}_I$ and a non-critical triangular decomposition $I = I_1 \cap \cdots \cap I_h$, one can compute the splitting of $\alpha$ in less than $\mathsf{A}_s(I)$ operations in $\mathbb{K}$. We denote `Split` such an algorithm.

**Theorem 7.** *If $I$ is defined by a triangular set $(P_1, \cdots, P_s)$, denoting $d_i = \deg_{Z_i}(P_i)$ and assuming $s$ to be constant, one can take $\mathsf{A}_s(I) \in \mathcal{O}^{\sim}(d_1 \cdots d_s)$*

*Proof.* This is a special case of the main result of [9], namely Theorem 8.1 therein. □

**Proposition 13.** *Let $a, b \in \mathbb{K}_I[Y]$ with $d_Y(a), \deg(b) \leq d$ and $I = (P_1, \cdots, P_s)$. Assuming $s$ to be constant, one can compute the extended greatest common divisor of $a$ and $b$ in less than $\mathcal{O}^{\sim}(d \cdot d_1 \cdots d_s)$ operations over $\mathbb{K}$.*

*Proof.* This is [9, Proposition 4.1]. □

**Splitting all coefficients of a polynomial.** We now consider a few additional results needed in our context, starting with the decomposition of a triangular set induced by checking regularity of all coefficients of a given polynomial.

**Lemma 12.** *There exists an algorithm `ReducePol` that, given $I = (Q, P)$ and $H \in \mathbb{K}_I[X,Y]$, returns a collection $\{(I_k, H_k)_k\}$ such that $I = \cap_k I_k$ is a non critical triangular decomposition and the polynomials $H_k = H \mod I_k$ are regular over $I_k$. This algorithm performs at most $\mathcal{O}^{\sim}(d_X(H)\, d_Y(H)\, d_P\, d_Q)$ operations over $\mathbb{K}$.*

*Proof.* The idea is similar to the one used in [9, Algorithm monic]. We just run a loop on the coefficients of $H$. We begin by splitting the coefficient according to the decomposition of $I$ found so far. For each element of the decomposition, we test the regularity of the coefficient using gcd computation. This gives us a new (possibly critical) decomposition of $I$. We run Algorithm `removeCriticalPairs` on it. At the end, we split $H$ according to the found decomposition. Complexity follows from Theorem 7 and Proposition 13. □

**Square-free decomposition above $\mathbb{K}_I$.** We say that a monic polynomial $P \in \mathbb{K}_I[Y]$ is square-free if the ideal $I + (P)$ is radical. We say that $P = \prod_i P_i^{n_i}$ is the square-free factorisation of $P$ over $\mathbb{K}_I$ if the $P_i$ are coprime square-free polynomials in $\mathbb{K}_I[Y]$ and $n_i < n_{i+1}$ for all $i$. We will use the following result:

**Proposition 14.** *Let $\mathbb{K}$ be a field of characteristic $p$ satisfying $p = 0$ or $p > d$. There exists an algorithm* `SQR-Free` *that, given a bivariate triangular set $I = (Q, P)$ and $\phi \in \mathbb{K}_I[Y]$ monic of degree $d$, computes a set $\{(I_k, (\phi_{k,l}, M_{k,l})_l)_k\}$ such that $I = \cap_k I_k$ is a non critical triangular decomposition of $I$ and, denoting $\phi_k := \phi \mod I_k$, $\phi_k = \prod_l \phi_{k,l}^{M_{k,l}}$ is the square-free factorisation of $\phi_k \in \mathbb{K}_{I_k}[Y]$. It takes less than $\mathcal{O}(d\, d_I)$ operations over $\mathbb{K}$.*

*Proof.* We just need to compute successive gcds and euclidean divisions, using Yun's algorithm [15, Algorithm 14.21, page 395] (this result is in characteristic 0, but works in positive characteristic when $p > d$). Each gcd computation is Proposition 13, and we just need to add splitting steps (if needed) of the ideal $I$ in between two calls. The complexity follows by using Proposition 13 in the proof of [15, Theorem 14.23, page 396], since there are less than $d$ calls to the algorithm `removeCriticalPairs`. □

**Computation of a primitive element.** We conclude this section by a brief description of the algorithm `PrimElt`. This is the same algorithm than the one described in Proposition 5, with additional attention on splittings. Indeed, in our context, we will need to compute primitive element above $\mathbb{K}_Q$ for some squarefree polynomial $Q$.

**Proposition 15.** *Assuming there is at least $d^2$ elements in $\mathbb{K}$, there exists a Las-Vegas algorithm that, given $\phi$ squarefree in $\mathbb{K}_I[Z_3]$ for some triangular set $I = (Q, P)$, compute a set $(Q_k, P'_k, \psi_k)$ of squarefree polynomials over $\mathbb{K}_{Q_k}$ with $d = \deg(P'_k) = d_P \deg_{Z_3}(\phi)$, $Q = \prod_k Q_k$ and such that $\psi_k$ is an isomorphism between $\mathbb{K}_{I_k}$ and $\mathbb{K}_{I'_k}$ (denoting $I_k = (Q_k, P \mod Q_k, \phi \mod Q_k)$ and $I'_k = (Q_k, P'_k)$) in less than $\mathcal{O}(d^{\frac{\omega+1}{2}} d_Q)$ operations over $\mathbb{K}$. We call* `PrimElt` *such an algorithm. Given $H \in \mathbb{K}_{I_k}[X, Y]$, one can compute $\psi_k(H_k) \in \mathbb{K}_{I'_k}[X, Y]$ in less than $\mathcal{O}(d_X(H)\, d_Y(H)\, d_P\, d\, d_{Q_k})$.*

*Proof.* In order to compute a primitive element and the associated isomorphism, we follow the Las Vegas algorithm[3] given in [27, Section 2.2]. This involves trace computation of the monomial basis, that is reduced to polynomial multiplication thanks to [20, Proposition 8] ; it takes $\mathcal{O}(\mathsf{M}(d\, d_Q))$ operations in $\mathbb{K}$. Then, picking a random element $A$, power projection is used to compute $2\,d$ traces of powers of $A$ ; using methodes based on [29], this involves only polynomial, transposed polynomial and matrix multiplications for a total in $\mathcal{O}(d^{\frac{\omega+1}{2}} \mathsf{M}(d_Q))$ operations in $\mathbb{K}$. Finally, the primitive element can be deduced via Newton's method in $\mathcal{O}(\mathsf{M}(d\, d_Q))$ operations ; testing its squarefreeness involves gcd over $\mathbb{K}_Q$, thus possible decomposition of $Q$. It takes less than $\mathcal{O}(d\, d_Q)$ operations over $\mathbb{K}$ from Proposition 13. Finally, one just need to use algorithm `removeCriticalPairs` when needed to deal with triangular decomposition that appear. This fits in our bound from Theorem 7.

For each $\psi_k$ computation, we need to compute $d$ additional traces ; this is once again power projection. Finally, one has to solve a linear system defined by a Hankel matrix (see [29, Proof of Theorem 5]). This can be solved using algorithm described in [4], that reduces the problem to extended gcd computation. This step thus involves potential decomposition of $Q$, and can be done once again in less than $\mathcal{O}(d\, d_Q)$ operations over $\mathbb{K}$.

Finally, rewriting the coefficient of a polynomial $H_k \in \mathbb{K}_{I_k}[X, Y]$ can be done in $\mathcal{O}(d_X(H_k)\, d_Y(H_k)\, d_P\, d\, d_{Q_k})$ using e.g. Horner's scheme [24, Section 5.1.3, page 209]. □

*Remark* 9. Computation of primitive element enables us to keep a constant number of variables (namely at most two) for the triangular sets we are using during the whole algorithm. Nevertheless, this complexity result does not enable to work with univariate triangular sets ; indeed, the degree $d_P$ of the current extension we are considering is powered up to $d_p^{\frac{\omega+1}{2}}$. Working with univariate extensions would lead to a bound in $(d_Q\, d_P)^{\frac{\omega+1}{2}}$, that can be $D^{\omega+1}$ when the factor $Q$ of the resultant has high degree (see Section 6). As $\omega > 2$, we could not achieve our aimed bounds. Moreover, $Q$ and $P$ do not provide the same geometrical information ; it is thus interesting to distinguish them.

**Extending [18, Algorithm 1] to the D5 context.**

**Proposition 16.** *Given $G, H \in \mathbb{K}_I[X, Y]$ with $I = (P, Q)$ and whose degrees in $Y$ are bounded by $d$, one can compute a set $(I_k, G_k, H_k, U_k, V_k, \eta_k)_k$ such that:*

- *$I = \cap_k I_k$ is a non critical decomposition of $I$,*
- *$G_k = G \mod I_k$ and $H_k = H \mod I_k$,*

[3] here the assumption on the number of elements of $\mathbb{K}$ is used

- *For each $k$, $U_k \cdot G_k + V_k \cdot H_k = X^{\eta_k} \mod X^{\eta_k+1}$ where $\eta_k \in \mathbb{N}$ is minimal such that $(G_k, H_k) \cap (X^{\eta_k}) = (X^{\eta_k})$.*

*This can be done in less than $\tilde{\mathcal{O}}(d\,\eta)$ arithmetic operations, with $\eta = \max_k \eta_k$.*

*Proof (sketch).* Algorithm given in [18] is similar to the fast-gcd algorithm: the two key ingredients in there are polynomial multiplication and euclidean division. As a consequence, one can adapt this algorithm as done in [9] for the fast-gcd algorithm. Writting that properly would require a lot of space ; as this is not a key point of this paper, we skip it. □

### 5.3 Computing polygon datas in the D5 context.

We now present the main algorithm where we have to deal with dynamic evaluation: the computation of the Newton polyon and associated datas, namely squarefree decomposition of characteristic polynomials. These computations are grouped in algorithm `Polygon-Data` below. Given $H \in \mathbb{K}_I[X,Y]$ known with precision $n$, it returns a list $\{(I_i, H_i, \Delta_{ij}, \phi_{ijk}, M_{ijk})\}$ where:

- $I = \cap I_i$ is a non critical triangular decomposition and the non zero coefficients of $H_i := H \mod I_i$ are invertible,
- $\{\Delta_{ij}\}_j = \mathcal{N}_n(H_i)$ is the set of edges of $H_i$ which can be computed from $\lceil H_i \rceil^n$ ; see Definition 8,
- $\phi_{\Delta_{ij}} = \prod_k \phi_{ijk}^{M_{ijk}}$ is the squarefree factorisation of $\phi_{\Delta_{ij}}$ the characteristic polynomial of $H_i$ associated to $\Delta_{ij}$.

**Algorithm:** `Polygon-Data`$(H, I, n)$

**Input:** $I$ a triangular set and $H \in \mathbb{K}_I[X,Y]$ a bivariate polynomial known modulo $X^{n+1}$. We assume $n > 0$ and $d_Y(H) > 0$.

**Output:** A list $\{(I_i, H_i, \Delta_{ij}, \phi_{ijk}, M_{ijk})\}$ as explained above.

```
1 foreach (H_i, I_i) in ReducePol(H, I) do
2     {Δ_ij}_{j=1,...,s} ← N_n(H_i) ;          // Coefficients of H_i are not zero divisors
3     for j = 1, ..., s do
4         {I_i^l, φ_ijk^l, M_ijk^l} ← SQR-Free(φ_{Δ_ij}, I_i)
5 {I'_m}_m ← removeCriticalPairs({I_i^l}_{i,l});
6 {H'_m}_m ← Split(H_i, {I_i^l}_{i,l}, {I'_m}_m);
7 foreach i, j, k do
8     {φ'_mjk}_mjk ← Split(φ_ijk^l, {I_i^l}_l, {I'_m}_m);   // assuming the good subset {I'_m}_m is taken
9 return {(I'_m, H'_m, Δ_{i(m)j}, φ_mjk, M_{i(m)jk})}_{m,j,k};   // i(m) is the correct i associated to m
```

**Proposition 17.** *Algorithm* `Polygon-Data` *works as expected, and assuming $I = (P, Q)$ represents a bivariate triangular set, it takes less than $\tilde{\mathcal{O}}(d_X(H)\, d_Y(H)\, d_I)$*

*Proof.* Exacteness and complexity are straightforward from Proposition 14 and Theorem 7, using $\sum_{j,k,l} \deg(\phi_{ijk}^l) \leq d_Y(H)$ for any $i$ and $\sum_{i,l} \deg(I_i^l) = \sum_m \deg(I'_m) = \sum_i \deg(I_i) = d_I$. □

### 5.4 Computing half Puiseux series using dynamic evaluation.

We finally provide the D5 variant `Half-RNP3` of algorithm `Half-RNP`: here we replace univariate factorisation by square-free factorisation and use dynamic evaluation to deal with potential zero divisors.

In order to compute also the RPEs of $F$ above the roots of any factor $Q$ of the resultant, we are led to consider $I = (Q, P)$ instead of $P$ as an input for `Half-RNP3`. More precisely, the input is a set $F, I, n, \pi$ such that:

- $I = (Q, P)$ is a bivariate triangular set over $\mathbb{K}$ ($P = Z_2$ for the initial call ; $Q = Z_1$ is admitted)
- $n \in \mathbb{N}$ is the truncation order we will use for the powers of $X$ during the algorithm,
- $F \in \mathbb{K}_I[X,Y]$ separable and monic in $Y$ with $d_Y > 0$ and known with precision $n$

- $\pi$ a pseudo-parametrisation defined by previous computations (we take $\pi = (X, Y)$ for the initial call).

The output is a set $\{I_i, \mathcal{R}_i\}_i$ such that:

- $I = \cap_i I_i$ is a non critial decomposition,
- $\mathcal{R}_i = \{R_{ij}\}$ is a set of D5-RPE's of $F_i := F \mod I_i$ which satisfy $n - v_{ij} \geq r_{ij}$ and which are given with precision at least $(n - v_{ij})/e_{ij} \geq r_{ij}/e_{ij} \geq 0$,

where we let $v_{ij} := v_X(\partial_Y F_i(S))$, with $S$ any Puiseux series associated to $R_{ij}$.

**Algorithm:** `Half-RNP3`$(F, I, n, \pi)$

$(I_i, H_i, \Delta_i, \phi_i, M_i)_i \leftarrow$ `Polygon-Data`$(F, I, n)$;
$\{\pi_i\}_i \leftarrow$ `Split`$(\pi, \{I_i\}_i)$; `// taking only once each different` $I_i$
**forall** $i$ **do**
  **if** $M_i = d_Y(H_i)$ **then**
    $\pi'_i \leftarrow \lceil \pi_i(X, Y - B_i) \rceil^n$; `//` $B_i :=$`Coeff`$(H_i, Y^{d_Y(H_i)-1})/d_Y(H_i)$
    **if** $M_i = 1$ **then** $\{I_{i11}, \mathcal{R}_{i11}\} \leftarrow \{(I_i, \pi'_i(T, 0))\}$;
    **else**
      $\{I_{i1k}, \mathcal{R}_{i1k}\}_k \leftarrow$ `Half-RNP3`$(\lceil H_i(X, Y - B_i) \rceil^n, I_i, n, \pi'_i)$
  **else**
    **if** $\deg(\phi_i) = 1$ **then** $\xi_{i1}, I_{i1}, H_{i1}, \pi_{i1}, s_i = -\phi_i(0), I_i, H_i, \pi_i, 1$;
    **else** `// computing a primitive representation`
      $\{I_{ij}, \Psi_{ij}\}_j \leftarrow$ `PrimElt`$(I_i, \phi_i)$ ; `//` $\Psi_{ij} : \mathbb{K}_{Q_{i,j}(Z_1), P_{i,j}(Z_1,Z_2), \phi_{i,j}(Z_1,Z_2,Z)} \to \mathbb{K}_{Q_{i,j}(W_1), P'_{i,j}(W_2)}$
      $\{H'_{ij}\}_j \leftarrow$ `Split`$(H_i, \{I_{ij}\}_j)$ ; $\{\pi'_{ij}\}_j \leftarrow$ `Split`$(\pi_i, \{I_{ij}\}_j)$;
      **forall** $j$ **do** $\xi_{ij}, H_{ij}, \pi_{ij} \leftarrow \Psi_{ij}(Z), \Psi_{ij}(H'_{ij}), \Psi_{ij}(\pi'_{ij})$ ;
    **forall** $j$ **do** `//` $\Delta_i$ `belongs to` $m_i\, a + q_i\, b = l_i$ ; $u_i, v_i =$ `Bézout`$(m_i, q_i)$
      $\pi''_{ij} \leftarrow \pi'_{ij}(\xi_{ij}^{v_i} X^{q_i}, X^{m_i}(Y + \xi_{ij}^{u_i}))$ $\mod I_{ij}$;
      $H''_{ij} \leftarrow \lceil H_{ij}(\xi_{ij}^{v_i} X^{q_i}, X^{m_i}(Y + \xi_{ij}^{u_i}))\rceil^{n_i}$ $\mod I_{ij}$ ; `//` $n_i = q_i\, n - l_i$
      $H'''_{ij} \leftarrow$`WPT`$(H''_{ij}, n_i)$;
      $\{I_{ijk}, \mathcal{R}_{ijk}\}_k \leftarrow$`Half-RNP3`$(H'''_{ij}, I_{ij}, n_i, \pi''_{ij})$; `// no recursive call if` $n_i \leq 0$[4]
$\{I'_l\}_l \leftarrow$ `removeCriticalPairs`$(\{I_{ijk}\}_{ijk})$;
$\mathcal{R} \leftarrow \{\}$;
**forall** $i, j, k$ **do** $\mathcal{R} \leftarrow \mathcal{R} \cup$ `Split`$(\mathcal{R}_{ijk}, \{I'_l\}_l)$; `// taking the subset of` $\{I'_l\}_l$ `refining` $I_{ijk}$
**return** $\mathcal{R}$; `// assuming each element of` $\mathcal{R}$ `is coupled to its associated` $I'_l$

We refer to the field version `Half-RNP` for all notations which are not specified here.

**Proposition 18.** *Let $Q \in \mathbb{K}[Z]$ be a square-free polynomial and let $F \in \mathbb{K}_Q[X, Y]$ be a polynomial monic and separable in $Y$. The function call* `Half-RNP3`$(F, (Q, Z), n, (X, Y))$ *returns a correct answer with at most $\mathcal{O}^{\sim}(d_Q\, n\, {d_Y}^2))$ operations over $\mathbb{K}$.*

*Proof.* The proof of this results is an adaptation of the proof of Proposition 7 to the D5 context, using Theorem 7 and Propositions 17 and 15. □

## 5.5 Proof of Theorem 1.

We can finally conclude the proof of Theorem 1 by providing the D5 variants of algorithms `MonicRNP` and `RNP`, namely algorithms `MonicRNP3` and `RNP3` below.

**The monic case.** As in Section 4, we begin with the monic case. Therein, we assume that the `Hensel` algorithm is a D5 version: we first use Proposition 16, and then a loop on each output for the remaining of the `Hensel` procedure.

**Algorithm:** `MonicRNP3`$(F, Q, n)$

**Input:** $Q \in \mathbb{K}[Z]$ square-free, $F \in \mathbb{K}_Q[X, Y]$, separable and monic in $Y$ and $n \in \mathbb{N}$ big enough.

**Output:** $\{(Q_i, \mathcal{R}_i)\}$, with $Q = \prod Q_i$ and $\mathcal{R}_i$ a system of singular parts of D5-RPEs of $F \mod Q_i$ above 0.

$\eta \leftarrow \min(n, 6\,n/d_Y)$ ; $\mathcal{R} \leftarrow \{\}$;
$\{I_i, \mathcal{R}_i\}_i \leftarrow$ `Half-RNP3`$(F, (Q, Z_2), \eta, \pi)$; // $I_i = (Q_i, Z_2)$
$\{F_i\}_i \leftarrow$ `Split`$(F, \{Q_i\}_i)$;
**forall** $i$ **do**
  Keep in $\mathcal{R}_i$ the $R_{ij}$ such that $v_{ij} < \eta/3$; // These RPEs are known with precision $\geq 2\eta/3$
  **if** $\sum_{k=1}^{\lambda_i} e_{ik}\, f_{ik} = d_Y$ **then** $\mathcal{R} \leftarrow \mathcal{R} \cup \{Q_i, \mathcal{R}_i\}$ ; **continue**;
  $G_i \leftarrow$ `NormRPE`$(\mathcal{R}_i, 2\eta/3)$;
  $H_i \leftarrow$ `Quo`$(F_i, G_i, 2\eta/3)$; // no splitting since $G_i$ is monic
  $\{Q_{ij}, G_{ij}, H_{ij}\}_j \leftarrow$ `Hensel`$(F_i, G_i, H_i, n)$;
  **forall** $j$ **do**
    $\{(Q_{ijk}, \mathcal{R}_{ijk})\}_k \leftarrow$ `MonicRNP3`$(H_{ij}, Q_{ij}, n, \pi)$;
  $\{\mathcal{R}'_{ijk}\} \leftarrow$ `Split`$(\mathcal{R}_i, \{Q_{ijk}\}_{j,k})$;
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(Q_{ijk}, \mathcal{R}_{ijk} \cup \mathcal{R}'_{ijk})_{j,k}\}$;
**return** $\mathcal{R}$

Recall the notations $R_F = \mathrm{Res}_Y(F, F_Y)$ and $v = v_X(R_F)$. We obtain the following result:

**Proposition 19.** *Assuming that $n \geq v$ and that the trailing coefficient of $R_F$ is not a zero divisor in $\mathbb{K}_Q$, a function call* `MonicRNP3`$(F, Q, n)$ *returns a correct answer in less than $\mathcal{O}(d_Q\, d_Y\, n)$ operation overs $\mathbb{K}$.*

*Proof.* The assumption on the trailing coefficient of the resultant of $F$ is needed only to ensure that the truncation bound $v$ is enough over all factors of $Q$. Otherwise, this is just an adaptation of the proof of Proposition 9 to the D5 context, using Propositions 18 and 16, together with Theorem 7 once again (subroutine `Quo` is used only with monic polynomials, and the remaining operations do not include any division). □

**The general case.** Algorithm `RNP3` below computes a system of singular part (at least) of D5-RPEs of any primitive polynomial $F$ above the roots of any square-free factor $Q$ of the resultant $R_F$. We follow the same strategy as in Algorithm `RNP`, but we take care of triangular decompositions due to division by zero divisors. In particular, we assume that algorithm `Monic` is a D5 version (some divisions occur while computing $F_\infty$). Also, inversion of the RPEs of $F'_\infty$ can lead to some triangular decompositions. However, we do not detail these further splittings for readibility.

**Algorithm:** `RNP3`$(F, Q, n)$

**Input:** $Q \in \mathbb{K}[Z_1]$ squarefree, $F \in \mathbb{K}[X, Y]$, separable in $Y$ with $d_Y > 0$ and $n \in \mathbb{N}$ big enough.

**Output:** A system of singular parts (at least) of D5-RPEs of $F$ above the roots of $Q$

$\mathcal{R} \leftarrow \{\}$ ; $F' \leftarrow \lceil F(X + Z_1, Y) \mod Q \rceil^n$; // thus $F' \in \mathbb{K}_Q[X, Y]$
$\{Q_i, F_{i,0}, F_{i,\infty}\}_i \leftarrow$ `Monic`$(F', n)$;
**forall** $i$ **do**
  $F'_{i,\infty} \leftarrow Y^{d_Y(F_{i,\infty})} F_{i,\infty}(X, 1/Y)$;
  $\{Q_{ij}, \mathcal{R}_{ij}\}_j \leftarrow$ `MonicRNP3`$(F_{i,0}, Q_i, n)$;
  $\{Q'_{ik}, \mathcal{R}'_{ik}\}_k \leftarrow$ `MonicRNP3`$(F'_{i,\infty}, Q_i, n)$;
  **forall** $k$ **do**
    Inverse the second element of each $R \in \mathcal{R}'_{ik}$; // Split $\{Q'_{ik}, \mathcal{R}'_{ik}\}$ if required
  $\{Q''_{il}\}_l \leftarrow$ `removeCriticalPairs`$(\{Q_{ij}\}_j \cup \{Q'_{ik}\}_k)$;
  **forall** $k, j$ **do**
    $\mathcal{R} \leftarrow \mathcal{R} \cup$ `Split`$(\mathcal{R}_{ij}, \{Q''_{il}\}_l) \cup$ `Split`$(\mathcal{R}'_{ik}, \{Q''_{il}\}_l)$;
**return** $\mathcal{R}$; // assuming elements of $\mathcal{R}$ with the same $Q''_{il}$ are grouped together

**Proposition 20.** *Assuming that $Q$ is a square-free factor of $R_F$ with multiplicity $\upsilon_Q \leq n$, a function call* `RNP3`$(F, Q, n)$ *returns the correct answer in less than $\mathcal{O}\tilde{}(d_Q\, d_Y\, n)$ operation overs $\mathbb{K}$.*

*Proof.* The correctness follows from Proposition 11 and Proposition 19 (the trailing coefficient of the resultant of $F_{i,0}$ and $F_{i,\infty}$ is not a zero divisor by construction). The complexity follows from Proposition 3, Theorem 7 and Proposition 19, combined with the relation $d_Y(F_{i,0}) + d_Y(F_{i,\infty}) = d_Y$ and $\sum_i \deg(Q_i) = d_Q$. □

*Proof of Theorem 1.* The algorithm mentionned in Theorem 1 is Algorithm `RNP3`, run with parameters $Q = Z_1$ and $n = \upsilon$. The computation of $\upsilon$ can be done via [18, Algorithm 1] in the aimed bound. Note that as we consider the special case $Q = Z_1$, $F$ has coefficients over a field and this operation does not involve any dynamic evaluation. The function call `RNP3`$(F, Z_1, \upsilon)$ fits in the aimed complexity thanks to Proposition 20. □

## 6 Desingularisation and genus of plane curves: proof of Theorem 2 and corollaries.

It's now straightforward to compute a system of singular parts of D5 rational Puiseux expansions above all critical points. We include the RPEs of $F$ above $X = \infty$, defined as RPEs above $X = 0$ of the reciprocal polynomial $F' := X^{d_X} F(X^{-1}, Y)$. We denote by $\upsilon_\infty := \upsilon_X(R_{F'})$, equal to $d_X\,(2\,d_Y - 1) - \deg(R_F)$.

**Definition 21.** Let $F \in \mathbb{K}[X, Y]$ be a separable polynomial over a field $\mathbb{K}$. A D5-desingularisation of $F$ over $\mathbb{K}$ is a collection $\{(\mathcal{R}_1, Q_1), \ldots, (\mathcal{R}_s, Q_s), \mathcal{R}_\infty\}$ such that:

- $Q_k \in \mathbb{K}[X]$ are pairwise coprime square-free polynomials such that $R_F = \prod_{k=1}^{s} Q_k^{\upsilon_k}$, $\upsilon_k \in \mathbb{N}^*$;
- $\mathcal{R}_k$ is a system of singular parts (at least) of D5-RPEs of $F$ above the roots of $Q_k$;
- $\mathcal{R}_\infty$ is a system of singular parts (at least) of D5-RPEs of $F$ above $X = \infty$.

Note that we can deduce from a D5-desingularisation of $F$ the singular part of the RPE's of $F$ above any roots of the resultant $R_F$. Also, note that we allow equality $\upsilon_k = \upsilon_l$ for $k \neq l$ so that the factorisation of the resultant $R_F$ does not necessarily coincides with its square-free factorisation. We obtain the following algorithm:

**Algorithm:** `Desingularise`$(F)$
**Input:** $F \in \mathbb{K}[X, Y]$ separable and primitive in $Y$, with $d_Y > 0$.
**Output:** The D5-desingularisation of $F$ over $\mathbb{K}$

$\mathcal{R} \leftarrow \{\}$;
**forall** $(Q, n) \in$ `SQR-Free`$(R_F)$ **do**
  $\mathcal{R} \leftarrow \mathcal{R} \cup$ `RNP3`$(F, Q, n)$;
$n \leftarrow d_X\,(2\,d_Y - 1) - \deg(R_F)$;
**if** $n > 0$ **then**
  $F' \leftarrow \lceil X^{d_X} F(X^{-1}, Y) \rceil^n$;
  $\mathcal{R} \leftarrow \mathcal{R} \cup$ `RNP3`$(F', Z, n)$
**return** $\mathcal{R}$

**Proposition 21.** *Algorithm* `Desingularise`$(F)$ *is correct and takes less than $\mathcal{O}\tilde{}(d_X\, {d_Y}^2)$ operations over $\mathbb{K}$.*

*Proof.* Correctness is straightforward from Proposition 20. The computation of the resultant $R_F$ fits in the aimed bound [15, Cor. 11.21]. The complexity is then straightforward from Proposition 20 combined with the classical formula $\sum_k \deg(Q_k)\upsilon_k + \upsilon_\infty = \deg(R_F) + \upsilon_\infty = d_X\,(2\,d_Y - 1)$. □

*Proof of Theorem 2.* Follows immediately from Proposition 21. □

**Computing the genus of plane curves: proof of Corollary 1.** Let $\{(Q_k, \mathcal{R}_k)\}$ be a D5-desingularisation of $F$. Since the coefficients of the D5-RPEs $R_{ki} \in \mathcal{R}_k$ are not zero divisors by construction, the ramification indices of all classical Puiseux series (i.e with coefficients in $\overline{\mathbb{K}}$) determined by $R_{ki}$ are equals. Assuming that $F$ is irreducible over $\overline{\mathbb{K}}$, the Riemann-Hurwitz formula thus determines the genus of the projective plane curve defined by $F$ as

$$g = 1 - d_Y + \frac{1}{2}\sum_k \deg(Q_k) \sum_{i=1}^{\rho_k} f_{ki}(e_{ki} - 1),$$

where $f_{ki}$ and $e_{ki}$ are respectively the residual degrees and ramification indices of the RPE $R_{ki}$. This proves Corollary 1. Corollary 2 and Corollary 3 then follow straightforwardly from the results in [23, 25] where the authors show that we can reduce $F$ modulo a well chosen small prime within the given bit complexities.

## 7 Factorization in $\mathbb{K}[[X]][Y]$. Proof of Theorem 3

We want to compute the irreducible analytic factors of $F$ in $\mathbb{K}[[X]][Y]$ with precision $X^N$ with $\mathcal{O}(d_Y(\upsilon+N))$ operations over $\mathbb{K}$ plus the cost of one univariate factorization of degree at most $d_Y$. We will use a dichotomic generalisation of Lemma 10. Given a family $F_1, \ldots, F_k \in \mathbb{K}[[X]][y]$, we denote by

$$\kappa = \kappa(F_1, \ldots, F_s) := \max_{I,J} \kappa(F_I, F_J),$$

the maximum of the lifting orders being taken over all disjoint subsets $I$, $J \subset \{1, \ldots, k\}$, with $F_I = \prod_{i\in I} F_i$.

**Proposition 22.** *Let $F \in \mathbb{K}[[X]][Y]$ be a degree $d$ separable polynomial. Suppose given a modular factorisation*

$$F \equiv uF_1 \cdots F_k \mod X^n, \quad n > 2\kappa \tag{1}$$

*where $u \in \mathbb{K}[X], u(0) \neq 0$ and where for all $i$, either $F_i$ is monic or its reciprocal polynomial $F_i'$ is monic. Then there exists uniquely determined analytic factors $F_1^*, \ldots, F_k^*$ such that*

$$F = u^* F_1^* \cdots F_k^*, \qquad F_i^* \equiv F_i \mod X^{n-\kappa} \text{ and } u^* \in \mathbb{K}[[X]] \text{ s.t. } u^* \equiv u \mod X^{n-\kappa},$$

*Moreover, starting from relation (1), we can compute the $F_i^*$ up to an arbitrary precision $N \geq n - \kappa$ within $\mathcal{O}(dN)$ operations over $\mathbb{K}$.*

*Proof.* Existence and unicity of the lifting follow from Lemma 10. Complexity follows from a dichotomic application of Lemma 10, as done for the multifactor Hensel lifting [15, Section 15.5]. □

*Remark* 10. This results improves [7, Lemma 4.1], where $\kappa$ is replaced by $\upsilon/2 \geq \kappa$. Note that if $\kappa = 0$, this is the classical multifactor Hensel lifting.

We only sketch a factorisation algorithm for proving Theorem 3. We will rather compute the analytic factors of $F$ with precision $\max(\upsilon/2, N)$.

**The monic case.** Suppose first that $F \in \mathbb{K}[X, Y]$ is a separable monic polynomial. The factorization algorithm follows from a straightforward modification of algorithm `MonicRNP3` :

1. We compute $\upsilon$ in the aimed bound.
2. By calling `Half-RNP3`$(F, (Z_1, Z_2), 6\upsilon/d_Y, (X, Y))$, we compute the set $\{R_1, \ldots, R_\lambda\}$ of all D5-RPEs of $F$ above 0 such that $v_i \leq 2\upsilon/_Y$ and given with precision $> 4\upsilon/d_Y$ in the aimed bound. We have $\sum_{i=1}^{\lambda} e_i f_i \geq d_Y/2$ by Corollary 5.
3. The $R_i$'s have coefficients in a product of fields $\mathbb{K}_{P_i}$, where $\sum \deg(P_i) = \sum f_i \leq d_Y$. Performing the univariate factorization of the $P_i$'s and splitting accordingly the $R_i$'s fit in the aimed bound. Hence, one can assume that the RPEs are defined over a field. Under such an assumption, they are in one-to-one correspondence with some irreducible analytic factors $F_1^*, \ldots, F_\lambda^*$ of $F$.

4. Let $F_i := \texttt{NormRPE}(R_i, 4v/d_Y)$. We have by construction $F = F_1 \cdots F_\lambda H \mod X^{4v/d_Y+1}$ for some polynomial $H$. The polynomials $F_i$ can be computed in the aimed bound, so does $H$ by euclidean division.

5. By Proposition 8, we have $\kappa(F_1, \ldots, F_\lambda, H) < \max v_i \leq 2v/d_Y$ and we deduce from Proposition 22 that we can lift the factorisation of step 4 and compute with at most $\mathcal{O}(d_Y v)$ operations a new factorisation

$$F = \tilde{F}_1 \cdots \tilde{F}_k \tilde{H} \mod X^{v+1},$$

where all factors coincide with the corresponding analytic factors $F_1^*, \ldots, F_k^*, H^*$ of $F$ up to precision $> v - 2v/d_Y$. Moreover, the $F_i^*$ are irreducible.

6. Previous equality forces $\tilde{H}$ to be separable. Moreover we have $d_Y(\tilde{H}) < d_Y(F)/2$ by Corollary 5. By applying recursively steps 2 to 5 on $\tilde{H}$ (keeping $v = v_F$ but replacing $d_Y$ with $d_Y(\tilde{H})$), we obtain within $\mathcal{O}(d_Y v)$ operations a factorisation

$$F = \tilde{F}_1 \cdots \tilde{F}_\rho \mod X^{v+1},$$

where the polynomials $\tilde{F}_i$ are in bijection with the irreducible analytic factors $F_i^*$ of $F$.

7. Since $\kappa(\tilde{F}_I, \tilde{F}_J) \leq v_X \mathrm{Res}_Y(\tilde{F}_I, \tilde{F}_J) \leq v/2$, Proposition 22 implies that $F_i^* = \tilde{F}_i \mod X^{v/2}$ for all $i$.

8. If $N \leq v/2$, we are done. Otherwise, we apply again Proposition 22 to lift the previous factorisation up to precision $X^{N+v/2}$ and compute in such a way the analytic factors of $F$ truncated with precision $N$ within $\mathcal{O}(d_Y N)$ operations.

**The non monic case.** If $F$ is not monic (but separable), we adapt now algorithm RNP3. We compute the triple $(u, F_0, F_\infty) := \texttt{Monic}(F, v)$ in the aimed bound. By construction, $F_0$ and the reciprocal polynomial $F'_\infty$ of $F_\infty$ are both monic. Moreover, they are separable (because known with precision $\geq v$), so we can compute their irreducible factors with precision $v$ in the aimed complexity bound by applying the monic case strategy. By considering the reciprocal factors of $F'_\infty$, we obtain in such a way a factorisation $F = u\tilde{F}_1 \cdots \tilde{F}_\rho \mod X^{v+1}$, where the $\tilde{F}_i$ are one-to-one with the irreducible factors of $F$. If necessary, we lift further this factorization to the desired precision as in step 8 above thanks to Proposition 22. Theorem 3 is proved. □

## References


[1] S. Abhyankar. *Algebraic Geometry for Scientists and Engineers*, volume 35 of *Mathematical surveys and monographs*. Amer. Math. Soc., 1990.

[2] J.-D. Bauch, E. Nart, and H. Stainsby. Complexity of the OM factorizations of polynomials over local fields. *LMS Journal of Computation and Mathematics*, 16:139–171, 2013.

[3] D. Bini and V. Y. Pan. *Polynomial and Matrix Computations*, volume 1 of *Progress in Theoretical Computer Science*. Birkhäuser, Saarbrücken, 1994.

[4] R. P. Brent, F. G. Gustavson, and D. Y. Yun. Fast solution of toeplitz systems of equations and computation of padé approximants. *Journal of Algorithms*, 1(3):259 – 295, 1980.

[5] E. Brieskorn and H. Knörrer. *Plane Algebraic Curves*. Birkhaüser, 1986.

[6] D. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1990.

[7] J. Cassel. *Local Fields*, volume 3 of *Student Texts*. LMS, 1986.

[8] C. Chevalley. *Introduction to the Theory of Algebraic Functions of One Variable*, volume 6 of *Mathematical Surveys*. AMS, 1951.

[9] X. Dahan, E. Schost, M. M. Maza, W. Wu, and Y. Xie. On the complexity of the D5 principle. *SIGSAM Bull.*, 39(3):97–98, 2005.

[10] J. Della Dora, C. Dicrescenzo, and D. Duval. About a new method for computing in algebraic number fields. In *EUROCAL 85*. Springer-Verlag LNCS 204, 1985.

[11] D. Duval. Diverses questions relatives au calcul formel avec des nombres algébriques. Université de Grenoble, Thèse d'État, 1987.

[12] D. Duval. Rational Puiseux expansions. *Compositio Math.*, 70(2):119–154, 1989.

[13] D. Duval and A. Poteaux. Death of marc rybowicz, aged 52. *ACM Commun. Comput. Algebra*, 50(4):191–191, Feb. 2017.

[14] M. Eichler. *Introduction to the Theory of Algebraic Numbers and Functions*. Academic Press, 1966.

[15] J. v. z. Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, New York, NY, USA, 3rd edition, 2013.

[16] F. Kako and T. Sasaki. Solving multivariate algebraic equations by Hensel construction. *Japan J. of Industrial and Applied Math.*, 16:257–285, 1999.

[17] H. T. Kung and J. F. Traub. All algebraic functions can be computed fast. *Journal of the Association for Computing Machinery*, 25(2):245–260, 1978.

[18] G. Moroz and É. Schost. A fast algorithm for computing the truncated resultant. In *ISSAC '16: Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, pages 1–8, New York, NY, USA, 2016. ACM.

[19] D. R. Musser. Multivariate polynomial factorization. *J. ACM*, 22(2):291–308, Apr. 1975.

[20] C. Pascal and E. Schost. Change of order for bivariate triangular sets. In *ISSAC'06*, pages 277–284. ACM, 2006.

[21] J. M. Peral. *Polígonos de newton de orden superior y aplicaciones aritméticas*. PhD thesis, Universitat de Barcelona, 1999.

[22] A. Poteaux. *Calcul de développements de Puiseux et application au calcul de groupe de monodromie d'une courbe algébrique plane*. PhD thesis, Université de Limoges, 2008.

[23] A. Poteaux and M. Rybowicz. Good reduction of puiseux series and complexity of the newton-puiseux algorithm over finite fields. In *ISSAC '08: Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, pages 239–246, New York, NY, USA, 2008. ACM.

[24] A. Poteaux and M. Rybowicz. Complexity bounds for the rational newton-puiseux algorithm over finite fields. *Applicable Algebra in Engineering, Communication and Computing*, 22:187–217, 2011. 10.1007/s00200-011-0144-6.

[25] A. Poteaux and M. Rybowicz. Good reduction of puiseux series and applications. *Journal of Symbolic Computation*, 47(1):32 – 63, 2012.

[26] A. Poteaux and M. Rybowicz. Improving complexity bounds for the computation of puiseux series over finite fields. In *Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '15, pages 299–306, New York, NY, USA, 2015. ACM.

[27] A. Poteaux and E. Schost. On the complexity of computing with zero-dimensional triangular sets. *Journal of Symbolic Computation*, 50(0):110 – 138, 2013.

[28] A. Schönage and V. Strassen. Schnelle multiplikation großer zahlen. *Computing 7*, pages 281–292, 1971.

[29] V. Shoup. Fast construction of irreducible polynomials over finite fields. *Journal of Symbolic Computation*, 17:371–391, 1994.

[30] R. J. Walker. *Algebraic Curves*. Springer-Verlag, 1950.

[31] M. Weimann. Bivariate factorization using a critical fiber. *Journal of Foundations of Computational Mathematics*, pages 1–45, 2016.