New Recombination Algorithms for Bivariate Polynomial Factorization Based on Hensel Lifting

Grégoire Lecerf

NEW RECOMBINATION ALGORITHMS FOR BIVARIATE POLYNOMIAL FACTORIZATION BASED ON HENSEL LIFTING

GRÉGOIRE LECERF

Abstract. We present new faster deterministic and probabilistic recombination algorithms to compute the irreducible decomposition of a bivariate polynomial via the classical Hensel lifting technique. For the dense bi-degree polynomial representation, the costs of our recombination algorithms are essentially sub-quadratic.

Introduction

Throughout this paper $\mathbb{K}$ denotes a commutative field of characteristic p, and F is a polynomial in $\mathbb{K}[x,y]$ of bi-degree (d_x,d_y), which means degree d_x in x and d_y in y. We are interested in the complexity of the computation of the irreducible factors $F_1,\dots,F_r$ of F together with their respective multiplicities $e_1,\dots,e_r$ for the *dense polynomial representation*, which means that a polynomial of bi-degree (d_x,d_y) is stored as the vector of its coefficients in the basis of the monomials of bi-degree at most (d_x,d_y). In this model, the *size* of F is precisely $(d_x+1)(d_y+1)$.

Main Results. For the cost analysis we use the *computation tree model* and the classical $\mathcal{O}$ and $\tilde{\mathcal{O}}$ (read "*soft Oh*") notation in the neighborhood of infinity as defined in [GG03, Chapter 25.7]: $f(d)\in\tilde{\mathcal{O}}(g(d))$ means that $f(d)\in g(d)(\log_2(3+g(d)))^{\mathcal{O}(1)}$. The factorization algorithms presented in this paper work for any characteristic but the coefficient field $\mathbb{K}$ must satisfy a few requirements: its cardinality must be sufficiently large, irreducible factorization in $\mathbb{K}[y]$ must be computable; when in positive characteristic, pth root extraction in $\mathbb{K}$ must be computable, and the $\mathbb{F}_p$-algebra structure of $\mathbb{K}$ must be effective. Herein $\mathbb{F}_p$ represents the prime subfield of $\mathbb{K}$. These requirements are detailed in Section 2.1.

For the sake of simplicity we will provide the reader only with a complete cost analysis in two cases: (a) in characteristic 0 in terms of the number of arithmetic operations in $\mathbb{K}$; (b) when $\mathbb{K}$ is the finite field $\mathbb{F}_{p^k}$ in terms of the number of arithmetic operations in $\mathbb{F}_p$.

The constant ω used below represents a feasible matrix multiplication exponent. We require that $2<\omega\le 3$ (see Section 2.1). The first theorem summarizes the cost of our deterministic reduction from two to one variables:

Theorem 1. *Assume that $\mathbb{K}$ has cardinality at least $2d_xd_y+\max(d_x,d_y)+1$. Then the computation of the irreducible decomposition $(F_1,e_1),\dots,(F_r,e_r)$ of F reduces to the computation of irreducible decompositions of polynomials in $\mathbb{K}[y]$ whose degree sum is at most d_x+d_y, plus*

a. *in characteristic 0, $\tilde{\mathcal{O}}((d_xd_y)^{(\omega+1)/2})$ arithmetic operations in $\mathbb{K}$;*

Date: Preliminary version of June 26, 2007.

2000 *Mathematics Subject Classification.* Primary 12Y05, 68W30; Secondary 11Y16, 12D05, 13P05.

Key words and phrases. Polynomial factorization, Hensel lifting.

This work was partly supported by the French Research Agency via the Gecko project (http://gecko.inria.fr).

b. *if* $\mathbb{K} := \mathbb{F}_{p^k}$, $\tilde{\mathcal{O}}(k(d_x d_y)^{(\omega+1)/2})$ *arithmetic operations in* $\mathbb{F}_p$.

Our second theorem yields a faster reduction but with a probability of failure. Therein the function R, defined in Section 2.1, is related to the cost of random subsets generation.

Theorem 2. *Assume that* $\mathbb{K}$ *has cardinality at least* $10 d_x d_y$. *Then the computation of the irreducible decomposition* $(F_1, e_1), \ldots, (F_r, e_r)$ *of* F *can be done with a probabilistic algorithm which performs irreducible factorizations of polynomials in* $\mathbb{K}[y]$ *whose degree sum is at most* $d_x + d_y$, *plus*

a. *in characteristic* 0, $\mathcal{O}((d_x d_y)^{1.5})$ *operations in* $\mathbb{K}$ *and* $\mathsf{R}(\mathcal{O}(d_x d_y))$;

b. *if* $\mathbb{K} := \mathbb{F}_{p^k}$, $\tilde{\mathcal{O}}(k(d_x d_y)^{1.5})$ *operations in* $\mathbb{F}_p$ *and* $\mathsf{R}(\mathcal{O}(d_x d_y))$.

The algorithm outputs either nothing or a correct result with a probability at least $1/2$.

The proofs of these two theorems are given at the end of Section 3.2. Informally speaking, if we consider that an arithmetic operation in $\mathbb{K}$ is not cheaper than k operations in $\mathbb{F}_p$, then the costs in parts (a) and (b) of the above theorems are of the same order of magnitude.

The mathematical ingredients of our algorithms are presented in the first section. Then the second section describes the core algorithms. The third section is devoted to the whole factorization algorithm.

Overview of the Algorithms. The algorithms underlying Theorems 1 and 2 both decompose into the two following stages:

1. *Reduction to the separable case.* Up to exchanging x and y we can assume that $d_y \leq d_x$. We compute the separable decomposition of F in y by means of the algorithm presented in [Lec07a].
2. *Factorization of the separable components.* In this stage F is assumed to be separable in y. Assuming that the cardinality of $\mathbb{K}$ is sufficiently large, a suitable shift of the variable x reduces the problem to the *normalized* case defined as follows:

$$(\mathrm{H}) \quad \begin{cases} \deg(F(0,y)) = d_y, \\ \mathrm{Res}\left(F(0,y), \frac{\partial F}{\partial y}(0,y)\right) \neq 0, \\ F \text{ is primitive when seen in } \mathbb{K}[x][y], \end{cases}$$

where $\mathrm{Res}(A, B)$ denotes the resultant of the two univariate polynomials A and B. In this case we can use the classical Hensel lifting strategy: from the irreducible factors $\mathfrak{F}_1(0,y), \ldots, \mathfrak{F}_s(0,y)$ of $F(0,y)$, we compute the irreducible *analytic decomposition* $\mathfrak{F}_1, \ldots, \mathfrak{F}_s$ of F in $\mathbb{K}[[x]][y]$ to a certain finite precision. Then we discover how these analytic factors *recombine* into $F_1, \ldots, F_r$.

The algorithms underlying Theorems 1 and 2 require a precision of the series only (x^{d_x+1}). In characteristic 0 or sufficiently large, these results are to be compared to ones of [Lec06] where the required precision is (x^{2d}) with d being the total degree of F. In positive characteristic, we improve the best previously known recombination algorithm designed in [BHKS04] that requires a precision $(x^{d_x(2d_y-1)+1})$ in the worst case [BHKS04, Theorem 5.2].

Related Works. The first polynomial time multivariate factorization algorithms are due to Kaltofen in the beginning of the eighties [Kal82b, Kal82c, Kal85]: they were derived from the algebraic approximant technique invented by A. K. Lenstra, H. W. Lenstra, and Lovász in 1982 to factor univariate polynomials over $\mathbb{Q}$ in polynomial time. Then Chistov, von zur Gathen, Grigoriev, Kaltofen, and A. K. Lenstra substantially contributed to this subject in order to reach polynomial time for all the usual fields. In 2003, Gao accomplished an important breakthrough with the design of a softly quadratic time probabilistic reduction of the factorization problem from two to one variable whenever the characteristic of the coefficient field is zero or sufficiently large [Gao03]. For more historical details we refer the reader to [Kal82a, Gat84, Kal90, Kal92, Zip93, Kal95, Kal03, GG03, Gao03, CG05].

The Hensel lifting strategy was popularized in computer algebra by Zassenhaus in 1969 in the context of factoring in $\mathbb{Q}[y]$ via a p-adic completion of $\mathbb{Q}$. For a long time, in the bivariate case, the cost of this approach has been remaining exponential in the number s of the analytic factors to recombine. Yet, as proved by Gao and Lauder, the cost of the recombination process is indeed softly linear in average [GL02] over finite fields, which explains the practical efficiency of this approach.

Of course a polynomial time factorization algorithm solves the recombination problem in polynomial time, but, to the best of our knowledge, the first polynomial time natural solution to this problem is due to van Hoeij for $\mathbb{Q}[y]$ [Hoe02], and to Belabas, van Hoeij, Klüners and Steel in the more general setting of global fields, that includes $\mathbb{K}(x)[y]$ [BHKS04]: the key idea is to recombine the logarithmic derivatives of the lifted factors. We refer to [Lec06] for a detailed historical survey on the recombination algorithms.

In [BLS$^+$04, Lec06, Lec07b] we presented recombination algorithms with a subquadratic cost in terms of the total degree, for the dense representation, and when the characteristic of $\mathbb{K}$ is 0 or sufficiently large. In the present paper we improve and generalize these results: our new algorithms require a smaller lifting precision and work whatever the characteristic of $\mathbb{K}$ is. In addition, instead of the total degree we deal with the bi-degree.

1. Reduction of the Recombination Problem to Linear Algebra

Throughout this section we assume that Hypothesis (H) holds. We explain how the recombination problem efficiently reduces to linear system solving. We separate the case when p is 0 or large enough to the specific case when $p > 0$. Yet we will see that the latter case makes use of the former case. The separation between these two cases is precisely given by the following condition:

(C) $\mathbb{K}$ has characteristic 0 or at least $d_x(2d_y - 1) + 1$.

1.1. Main Notation.

Let $c \in \mathbb{K}[x]$ (respectively, $c_i \in \mathbb{K}[x]$ for each $i \in \{1, \ldots, r\}$) denote the leading coefficient of F (respectively, of F_i) when seen in $\mathbb{K}[x][y]$. For convenience we assume that $c, c_1, \ldots, c_r$ are all monic (*i.e.* their leading coefficients equal 1), so that we can write $F = F_1 \cdots F_r$. Recall that $\mathfrak{F}_1, \ldots, \mathfrak{F}_s$ represent the monic irreducible factors of F seen in $\mathbb{K}[[x]][y]$. Then, to each $i \in \{1, \ldots, r\}$, we associate the vector $\mu_i \in \{0, 1\}^s$, uniquely defined by

$$F_i = c_i \prod_{j=1}^{s} \mathfrak{F}_j^{\mu_{i,j}}. \tag{1}$$

Since the μ_i have entries in $\{0, 1\}$ and have pairwise distinct supports, up to a unique permutation, they form a *reduced echelon basis*, which means that the $r \times s$ matrix whose ith row is μ_i is in the row-reduced echelon form. From now on we assume that $\mu_1, \ldots, \mu_r$ are actually ordered so that they form a reduced echelon basis. In this setting the recombination problem reformulates as follows: from the knowledge of the $\mathfrak{F}_i$ to a sufficient finite precision, compute all the vectors μ_i. Then it is classical that the knowledge of the μ_i suffices to recover $F_1, \ldots, F_r$ efficiently (see Algorithm 3 in Section 2.4).

If $A := \sum_{i,j\geq 0} a_{i,j}x^i y^j$ denotes a polynomial in $\mathbb{K}[[x]][y]$ then we write $\lceil A\rceil^l$ and $[A]_k^l$ respectively for the following projections to $\mathbb{K}[x,y]$:

$$\lceil A\rceil^l := \sum_{0\leq i\leq l-1,\ j\geq 0} a_{i,j}x^i y^j, \quad [A]_k^l := \sum_{k\leq i\leq l-1,\ j\geq 0} a_{i,j}x^i y^j.$$

We also introduce the following partial products:

$$\hat{F}_i := \prod_{j=1, j\neq i}^{r} F_j = \frac{F}{F_i} \text{ and } \hat{\mathfrak{F}}_i := c\prod_{j=1, j\neq i}^{s} \mathfrak{F}_j = \frac{F}{\mathfrak{F}_i}.$$

Finally, the central objects of our recombination process are the following:

$$\mathfrak{G}_i := \left\lceil \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial y}\right\rceil^{d_x+1}, \text{ for all } i\in\{1,\ldots,s\}.$$

The recombination algorithms will only make use of these objects, so that the only lifting to precision (x^{d_x+1}) is necessary.

Throughout this paper $\mathbb{F}$ denotes a subfield of $\mathbb{K}$, and the space of the bivariate polynomials over $\mathbb{K}$ of bi-degree at most (k,l) is written $\mathbb{K}[x,y]_{k,l}$. Our goal will be the computation of the following $\mathbb{F}$-vector space:

$$\mathcal{L}_{\mathbb{F}} := \left\{(\ell_1,\ldots,\ell_s)\in\mathbb{F}^s \mid \sum_{i=1}^{s}\ell_i\mathfrak{G}_i \in \left\langle \hat{F}_1\frac{\partial F_1}{\partial y},\ldots,\hat{F}_r\frac{\partial F_r}{\partial y}\right\rangle_{\mathbb{F}}\right\},$$

where $\left\langle \hat{F}_1\frac{\partial F_1}{\partial y},\ldots,\hat{F}_r\frac{\partial F_r}{\partial y}\right\rangle_{\mathbb{F}}$ represents the vector space spanned by the polynomials $\hat{F}_1\frac{\partial F_1}{\partial y},\ldots,\hat{F}_r\frac{\partial F_r}{\partial y}$ over $\mathbb{F}$. It turns out that the knowledge of $\mathcal{L}_{\mathbb{F}}$ solves the recombination problem:

Lemma 1. *$\mu_1,\ldots,\mu_r$ is the reduced echelon basis of $\mathcal{L}_{\mathbb{F}}$.*

Proof. Let $i\in\{1,\ldots,r\}$. By taking the logarithmic derivative in y and by multiplying by F both sides of (1), we obtain that:

$$\hat{F}_i\frac{\partial F_i}{\partial y} = \sum_{j=1}^{s}\mu_{i,j}\hat{\mathfrak{F}}_j\frac{\partial\mathfrak{F}_j}{\partial y}.$$

Since $\deg_y(F_1)+\cdots+\deg_y(F_s)=d_y$, $\deg_y\left(\frac{\partial F_i}{\partial y}\right)\leq\deg_y(F_i)-1$, $\deg_x(F_1)+\cdots+\deg_x(F_s)=d_x$, and $\deg_x\left(\frac{\partial F_i}{\partial y}\right)\leq\deg_x(F_i)$, we obtain $\hat{F}_i\frac{\partial F_i}{\partial y}\in\mathbb{K}[x,y]_{d_x,d_y-1}$, and deduce that:

$$\hat{F}_i\frac{\partial F_i}{\partial y} = \left\lceil\hat{F}_i\frac{\partial F_i}{\partial y}\right\rceil^{d_x+1} = \left\lceil\sum_{j=1}^{s}\mu_{i,j}\hat{\mathfrak{F}}_j\frac{\partial\mathfrak{F}_j}{\partial y}\right\rceil^{d_x+1} = \sum_{j=1}^{s}\mu_{i,j}\mathfrak{G}_j. \tag{2}$$

Since $\mu_i\in\{0,1\}^s\subseteq\mathbb{F}^s$, it follows that $\mu_i\in\mathcal{L}_{\mathbb{F}}$.

By Hypothesis (H), $\mathfrak{G}_1(0,y),\ldots,\mathfrak{G}_s(0,y)$ are linearly independent over $\mathbb{F}$, so are $\mathfrak{G}_1,\ldots,\mathfrak{G}_s$. By the same argument, $\hat{F}_1\frac{\partial F_1}{\partial y},\ldots,\hat{F}_r\frac{\partial F_r}{\partial y}$ are also linearly independent over $\mathbb{F}$. Therefore $\mathcal{L}_{\mathbb{F}}$ has dimension r, which concludes the proof. □

1.2. **Partial Fraction Decomposition.** Let $(\ell_1,\ldots,\ell_s)$ be a vector in $\mathbb{F}^s$, let $G := \sum_{i=1}^{s}\ell_i\mathfrak{G}_i$, and let $\bar{\mathbb{K}}$ denote the algebraic closure of $\mathbb{K}$. In order to construct a set of linear equations over $\mathbb{F}$ whose solution set is $\mathcal{L}_{\mathbb{F}}$, we consider the partial fraction decomposition of G/F in $\bar{\mathbb{K}}[[x]]$. Let $\phi_1,\ldots,\phi_{d_y}$ represent the roots of F in $\bar{\mathbb{K}}[[x]]$, and let $\rho_i := G(x,\phi_i)/\frac{\partial F}{\partial y}(x,\phi_i)$, for all $i\in\{1,\ldots,d_y\}$, so that we have:

$$\frac{G}{F} = \sum_{i=1}^{d_y}\frac{\rho_i}{y-\phi_i}. \tag{3}$$

This partial fraction decomposition is well defined over $\bar{\mathbb{K}}[[x]]$ thanks to Hypothesis (H).

Lemma 2. *If $(\ell_1,\dots,\ell_s) \in \mathcal{L}_{\mathbb{F}}$ then $\rho_1,\dots,\rho_{d_y}$ all belong to $\mathbb{F}$. Conversely, if $\rho_1,\dots,\rho_{d_y}$ all belong to $\bar{\mathbb{K}}$ then $(\ell_1,\dots,\ell_s) \in \mathcal{L}_{\mathbb{F}}$.*

Proof. By definition, if $(\ell_1,\dots,\ell_s) \in \mathcal{L}_{\mathbb{F}}$ then G is a $\mathbb{F}$-linear combination of the $\hat{F}_1\frac{\partial F_1}{\partial y},\dots,\hat{F}_r\frac{\partial F_r}{\partial y}$. Since, for all i, the residues of $\hat{F}_i\frac{\partial F_i}{\partial y}/F = \frac{\partial F_i}{\partial y}/F_i$ are all in $\{0,1\}$, we obtain that all the ρ_i belong to $\mathbb{F}$. Conversely, assume that all the ρ_i are in $\bar{\mathbb{K}}$. By substituting 0 for x in (3) and in the definition of G, we obtain

$$\frac{G(0,y)}{F(0,y)} = \sum_{i=1}^{d_y} \frac{\rho_i(0)}{y-\phi_i(0)} = \sum_{j=1}^{s} \ell_j \frac{\frac{\partial \mathfrak{F}_j}{\partial y}(0,y)}{\mathfrak{F}_j(0,y)},$$

so that Hypothesis (H) implies that $\rho_i = \rho_i(0) = \ell_j$ whenever $\mathfrak{F}_j(0,\phi_i(0)) = 0$, whence

$$G = \sum_{j=1}^{s} \ell_j \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial y}.$$

Now let i and j in $\{1,\dots,s\}$ be such that $\mathfrak{F}_i$ and $\mathfrak{F}_j$ both divide the same irreducible factor F_k of F for some $k \in \{1,\dots,r\}$. By construction, $\mathfrak{F}_i$ and $\mathfrak{F}_j$ respectively divide $\ell_i\frac{\partial F}{\partial y} - G$ and $\ell_j\frac{\partial F}{\partial y} - G$ in $\mathbb{K}[[x]][y]$. Since the two latter expressions are in $\mathbb{K}[x,y]$, they are multiple of F_k. It follows that $\mathfrak{F}_i$ divides $\ell_j\frac{\partial F}{\partial y} - G$ in $\mathbb{K}[[x]][y]$, hence that $\ell_i = \ell_j$ since $\frac{\partial F}{\partial y}$ is invertible modulo $\mathfrak{F}_i$ by Hypothesis (H). Therefore $(\ell_1,\dots,\ell_s)$ is a $\mathbb{F}$-linear combination of the μ_i and the conclusion thus follows from Lemma 1. □

1.3. **Characteristic Zero or Large Enough.** We carry on with the notation: $(\ell_1,\dots,\ell_s) \in \mathbb{F}^s$, $G := \sum_{i=1}^{s} \ell_i \mathfrak{G}_i$, and $\rho_1,\dots,\rho_{d_y}$ are the residues of G/F defined in (3). In characteristic 0, in order to ensure that all the ρ_i are in $\bar{\mathbb{K}}$, it suffices to ensure that all the ρ_i' are zero. Letting

$$\begin{aligned}\tilde{\mathsf{D}}:\quad & \mathbb{K}[x,y]_{d_x,d_y-1} \to \mathbb{K}[x,y]_{3d_x-1,3d_y-3}\\ & G \mapsto \left(\frac{\partial G}{\partial x}\frac{\partial F}{\partial y} - \frac{\partial G}{\partial y}\frac{\partial F}{\partial x}\right)\frac{\partial F}{\partial y} - \left(\frac{\partial^2 F}{\partial xy}\frac{\partial F}{\partial y} - \frac{\partial^2 F}{\partial y^2}\frac{\partial F}{\partial x}\right)G,\end{aligned}$$

it is classical that ρ_i' can be calculated as follows:

$$\begin{aligned}\rho_i' &= \frac{d}{dx}\left(\frac{G(x,\phi_i(x))}{\frac{\partial F}{\partial y}(x,\phi_i(x))}\right)\\ &= \frac{\frac{\partial G}{\partial x}(x,\phi_i(x)) + \frac{\partial G}{\partial y}(x,\phi_i(x))\phi_i'(x)}{\frac{\partial F}{\partial y}(x,\phi_i(x))}\\ &\quad - \frac{\frac{\partial^2 F}{\partial xy}(x,\phi_i(x)) + \frac{\partial^2 F}{\partial y^2}(x,\phi_i(x))\phi_i'(x)}{\frac{\partial F}{\partial y}(x,\phi_i(x))^2}G(x,\phi_i(x))\\ &= \frac{\tilde{\mathsf{D}}(G)(x,\phi_i(x))}{\frac{\partial F}{\partial y}(x,\phi_i(x))^3},\end{aligned}$$

since $\phi_i'(x) = -\frac{\partial F}{\partial x}(x,\phi_i(x))/\frac{\partial F}{\partial y}(x,\phi_i(x))$. Testing the vanishing of all the ρ_i' thus becomes equivalent to testing if F divides $\tilde{\mathsf{D}}(G)$ in $\mathbb{K}[[x]][y]$. Recall that the division is well-defined since c is invertible in $\mathbb{K}[[x]]$ by Hypothesis (H). The following lemma will lead us to an efficient algorithm for the latter test.

Lemma 3. *Let Q and R respectively denote the quotient and remainder of $\tilde{\mathsf{D}}(G)$ divided by F in $\mathbb{K}[[x]][y]$, so that we have $\tilde{\mathsf{D}}(G) = QF + R$, with $\deg_y(R) \leq d_y - 1$. Then F divides $\tilde{\mathsf{D}}(G)$ in $\mathbb{K}[[x]][y]$ if, and only if, $[Q]_{2d_x}^{3d_x} = \lceil R \rceil^{3d_x} = 0$.*

Proof. Let us assume that F divides $\tilde{\mathsf{D}}(G)$ in $\mathbb{K}[[x]][y]$. Then is it clear that $R = 0$ and $\tilde{\mathsf{D}}(G) = QF$ hold in $\mathbb{K}[[x]][y]$. Since $\tilde{\mathsf{D}}(G)$ and F are polynomials then $Q \in \mathbb{K}(x)[y]$ and $\tilde{\mathsf{D}}(G) = QF$ holds in $\mathbb{K}(x)[y]$. Let $a \in \mathbb{K}[x]$ and $A \in \mathbb{K}[x, y]$ be such that $Q = A/a$, so that we can write $a\tilde{\mathsf{D}}(G) = AF$ in $\mathbb{K}[x][y]$. Since F is primitive by Hypothesis (H), the classical Gauss lemma [Lan02, Chapter IV, Theorem 2.1] implies that a divides the content of A. It follows that Q actually belongs to $\mathbb{K}[x, y]$, whence $\deg_x(Q) \leq 2d_x - 1$.

Conversely, let us assume that $[Q]_{2d_x}^{3d_x} = \lceil R \rceil^{3d_x} = 0$. Then $\tilde{\mathsf{D}}(G) = \lceil Q \rceil^{2d_x} F$ holds in $K[[x]]/(x^{3d_x})[y]$. Since $\deg_x(\lceil Q \rceil^{2d_x} F) \leq 3d_x - 1$, we deduce that $\tilde{\mathsf{D}}(G) = \lceil Q \rceil^{2d_x} F$ holds in $\mathbb{K}[x, y]$, hence that F divides $\tilde{\mathsf{D}}(G)$ in $\mathbb{K}[[x]][y]$. □

For any $p > 0$ we write $\bar{\mathbb{K}}[[x^p]]$ for the series algebra in x^p over $\bar{\mathbb{K}}$. When $p = 0$, by convention we let $\bar{\mathbb{K}}[[x^0]] := \bar{\mathbb{K}}$. Lemma 3 motivates us to define the following map:

$$\begin{aligned} \mathsf{D}_{\mathbb{F}} : \quad & \mathbb{F}^s \to \mathbb{K}[x, y]_{d_x-1, 2d_y-3} \times \mathbb{K}[x, y]_{3d_x-1, d_y-1} \\ & (\ell_1, \ldots, \ell_s) \mapsto ([Q]_{2d_x}^{3d_x} / x^{2d_x}, \lceil R \rceil^{3d_x}). \end{aligned}$$

Proposition 1. *The inclusion $\langle \mu_1, \ldots, \mu_r \rangle_{\mathbb{F}} \subseteq \ker(\mathsf{D}_{\mathbb{F}})$ always holds. Conversely, if $(\ell_1, \ldots, \ell_s)$ belongs to $\ker(\mathsf{D}_{\mathbb{F}})$ then $\rho_1, \ldots, \rho_{d_y}$ all belong to $\bar{\mathbb{K}}[[x^p]]$. In addition, if (C) holds then $\mu_1, \ldots, \mu_r$ is the reduced echelon basis of $\ker(\mathsf{D}_{\mathbb{F}})$.*

Proof. Lemma 3 reformulates as follows: $(\ell_1, \ldots, \ell_s)$ belongs to $\ker(\mathsf{D}_{\mathbb{F}})$ if, and only if, $\rho_1, \ldots, \rho_{d_y}$ all belong to $\bar{\mathbb{K}}[[x^p]]$. If $(\ell_1, \ldots, \ell_s) = \mu_j$ then we have that $G = \hat{F}_j \frac{\partial F_j}{\partial y}$ by (2), hence that the ρ_j belong to $\{0, 1\}$, whence the inclusion $\langle \mu_1, \ldots, \mu_r \rangle_{\mathbb{F}} \subseteq \ker(\mathsf{D}_{\mathbb{F}})$.

Let $(\ell_1, \ldots, \ell_s) \in \ker(\mathsf{D}_{\mathbb{F}})$ and assume that (C) holds. In order to prove that $(\ell_1, \ldots, \ell_s) \in \langle \mu_1, \ldots, \mu_r \rangle_{\mathbb{F}}$, it is sufficient to prove that all the ρ_i belong $\bar{\mathbb{K}}$, by Lemmas 1 and 2. The case when $p = 0$ is immediate so that we can now assume that $p \geq d_x(2d_y - 1) + 1$. Here we could directly invoke [Gao03, Lemma 2.4] in order to conclude the proof, but, for completeness, let us briefly repeat the arguments. Let $i \in \{1, \ldots, d_y\}$ and let $A \in \bar{\mathbb{K}}[x, y]$ denote the unique irreducible factor of F in $\bar{\mathbb{K}}[x, y]$ such that $A(x, \phi_i)$=0. By construction the resultant

$$B := \mathrm{Res}_y \left(A, \rho_i(0) \frac{\partial F}{\partial y} - G \right) \in \bar{\mathbb{K}}[x]$$

has degree at most $d_x(2d_y - 1)$ and belongs to (x^p), hence it is zero since $p \geq \deg(B) + 1$. It follows that A divides $\rho_i(0) \frac{\partial F}{\partial y} - G$ in $\bar{\mathbb{K}}[[x]][y]$, whence $\rho_i(0) = G(x, \phi_i) / \frac{\partial F}{\partial y}(x, \phi_i) = \rho_i$. □

1.4. **Small Positive Characteristic.** Proposition 1 reduces the recombination problem to linear algebra under Hypothesis (C). Until the end of this section, we focus on the situation when (C) does not hold, and $\mathbb{F}$ exclusively denotes the prime field $\mathbb{F}_p$ of $\mathbb{K}$. We shall use the following $\mathbb{F}$-linear maps, respectively reminiscent of Berlekamp's and Niederreiter's univariate factorization algorithms (for instance, see [GG03, Chapter 14]):

$$\begin{aligned} \mathsf{B} : \quad & \mathbb{K}[x, y]_{d_x, d_y-1} \to \mathbb{K}(x)[y]/(F) \\ & G \mapsto G^p - \left(\frac{\partial F}{\partial y} \right)^{p-1} G \end{aligned} \qquad \begin{aligned} \mathsf{N} : \quad & \mathbb{K}[x, y]_{d_x, d_y-1} \to \mathbb{K}[x, y^p]_{pd_x, d_y-1} \\ & G \mapsto G^p + \frac{\partial^{p-1}}{\partial y^{p-1}} \left(F^{p-1} G \right). \end{aligned}$$

Here $\mathbb{K}[x, y^p]_{pd_x, d_y-1}$ is to be understood as the space of the polynomials in x and y^p of degrees at most pd_x in x and at most $d_y - 1$ in y^p. One can easily see that N is well defined since $\frac{\partial}{\partial y}\mathsf{N}(G) = 0$. For convenience, Appendix A gathers the classical properties needed in the sequel. As for D, we will be interested in the kernels of the following $\mathbb{F}$-linear maps:

$$\begin{aligned} \mathsf{B}_{\mathbb{F}} : \quad & \mathbb{F}^s \to \mathbb{K}(x)[y]/(F) \\ & (\ell_1, \ldots, \ell_s) \mapsto \mathsf{B}\left(\sum_{i=1}^{s} \ell_i \mathfrak{G}_i\right) \end{aligned} \qquad \begin{aligned} \mathsf{N}_{\mathbb{F}} : \quad & \mathbb{F}^s \to \mathbb{K}[x, y^p]_{pd_x, d_y-1} \\ & (\ell_1, \ldots, \ell_s) \mapsto \mathsf{N}\left(\sum_{i=1}^{s} \ell_i \mathfrak{G}_i\right). \end{aligned}$$

We carry on with the notation: $(\ell_1, \ldots, \ell_s)$ denotes a vector in $\mathbb{F}^s$, $G := \sum_{i=1}^{s} \ell_i \mathfrak{G}_i$, and $\rho_1, \ldots, \rho_{d_y}$ represent the residues of G/F as defined in (3).

Proposition 2. *$\mu_1, \ldots, \mu_r$ is the reduced echelon basis of* $\ker(\mathsf{B}_{\mathbb{F}}) = \ker(\mathsf{N}_{\mathbb{F}})$.

Proof. Proposition 11 of Appendix A applied with F seen in $\mathbb{K}(x)[y]$ and $\mathbb{E} := \bar{\mathbb{K}}((x))$ (the fraction field of $\bar{\mathbb{K}}[[x]]$) gives us that $\ker(\mathsf{B}_{\mathbb{F}}) = \ker(\mathsf{N}_{\mathbb{F}})$, and that $(\ell_1, \ldots, \ell_s)$ belongs to these kernels if, and only if, all the ρ_i belong to $\mathbb{F}$. The conclusion thus follows from Lemmas 1 and 2. □

Proposition 2 does not directly provide us with a satisfactory recombination algorithm because the linear system to be solved involves a number of equations that grows linearly with p. The key idea to cut down this dependency in p is to combine $\mathsf{N}_{\mathbb{F}}$ with $\mathsf{D}_{\mathbb{F}}$ as follows:

Lemma 4. *If* $(\ell_1, \ldots, \ell_s) \in \ker(\mathsf{D}_{\mathbb{F}})$ *then* $\mathsf{N}_{\mathbb{F}}(\ell_1, \ldots, \ell_s)$ *belongs to* $\mathbb{K}[x^p, y^p]_{d_x, d_y-1}$.

Proof. If $(\ell_1, \ldots, \ell_s) \in \ker(\mathsf{D}_{\mathbb{F}})$ then $\rho_1, \ldots, \rho_{d_y}$ all belong to $\bar{\mathbb{K}}[[x^p]]$ by Proposition 1. Therefore equality (8) of Appendix A.1 rewrites into

$$\mathsf{N}(G) = F^p \left(\sum_{i=1}^{d_y} \frac{\rho_i^p - \rho_i}{(y - \phi_i)^p}\right),$$

whence $\mathsf{N}_{\mathbb{F}}(\ell_1, \ldots, \ell_s) = \mathsf{N}(G) \in \mathbb{K}[x^p, y^p]_{d_x, d_y-1}$. □

Let $\Delta_y(x)$ represent the discriminant of F in y, that is $\mathrm{Res}_y\left(F(x, y), \frac{\partial F}{\partial y}(x, y)\right)$. We are now ready to present the test for the vanishing of $\mathsf{N}(G)$ and $\mathsf{B}(G)$ that will be used in the algorithms:

Proposition 3. *Let S be a set containing at least $d_x + 1$ points in $\bar{\mathbb{K}}$ and assume that $(\ell_1, \ldots, \ell_s) \in \ker(\mathsf{D}_{\mathbb{F}})$. Then the following equivalence holds:*

$$\mathsf{N}_{\mathbb{F}}(\ell_1, \ldots, \ell_s) = 0 \iff \forall a \in S, \ \mathsf{N}_{\mathbb{F}}(\ell_1, \ldots, \ell_s)(a, y) = 0.$$

In addition, if $c(a)\Delta_y(a) \neq 0$ for all $a \in S$, then the latter equivalence holds if we replace $\mathsf{N}_{\mathbb{F}}$ by $\mathsf{B}_{\mathbb{F}}$.

Proof. Since the map $z \mapsto z^p$ is injective in $\bar{\mathbb{K}}$, the vanishing of $\mathsf{N}(G)$ can be tested with only $d_x + 1$ distinct specializations of x by Lemma 4. By Proposition 11 of the appendix applied with F seen in $\mathbb{K}(x)[y]$, the equality $\mathsf{B}(G) = 0$ is equivalent to $\mathsf{N}(G) = 0$. Finally by the same proposition 11 applied with $F(a, y)$ seen in $\bar{\mathbb{K}}[y]$, the equality $\mathsf{B}(G)(a, y) = 0$ is equivalent to $\mathsf{N}(G)(a, y) = 0$, for any $a \in \bar{\mathbb{K}}$ satisfying $c(a)\Delta_y(a) \neq 0$. □

2. Recombination Algorithms

In this section, we present a deterministic recombination algorithm and a faster probabilistic one both derived from the results of the previous section. We still assume that the normalization Hypothesis (H) holds. The subfield $\mathbb{F}$ is set to be $\mathbb{K}$ if (C) holds and to be the prime field $\mathbb{F}_p$ of $\mathbb{K}$ otherwise.

2.1. **Complexity Model.** For our cost analysis, we use the *computation tree* model [BCS97, Chapter 4] from the *total complexity* point of view. Roughly speaking, this means that complexity estimates charge a constant cost for each arithmetic operation ($+$, $-$, $\times$, $\div$) and the equality test. Yet all the constants in the base fields (or rings) of the trees are thought to be freely at our disposal.

2.1.1. *Polynomial Arithmetic.* A univariate polynomial of degree d is thought to be represented as the vector of its coefficients of size $d+1$. For each integer d, we assume that we are given a computation tree that computes the products of two polynomials of degree at most d with at most $\mathsf{M}(d)$ operations, independently of the base ring. As in [GG03, Chapter 8.3], for any positive integers d_1 and d_2, we assume that M satisfies the following properties: $\mathsf{M}(d_1 d_2) \leq d_1^2 \mathsf{M}(d_2)$ and $\mathsf{M}(d_1)/d_1 \leq \mathsf{M}(d_2)/d_2$ if $d_1 \leq d_2$. In particular, this implies the *super-additivity* of M, that is $\mathsf{M}(d_1) + \mathsf{M}(d_2) \leq \mathsf{M}(d_1 + d_2)$.

During the cost analyzes we will appeal to the following classical results:

- The resultant and the extended greatest common divisor of two univariate polynomials of degree at most d over a field $\mathbb{K}$ can be computed with $\mathcal{O}(\mathsf{M}(d)\log(d))$ operations in $\mathbb{K}$ [GG03, Chapter 11].
- The product of r univariate polynomials $G_1, \dots, G_r$ over a field $\mathbb{K}$ whose degree sum is d takes $\mathcal{O}(\mathsf{M}(d)\log(r))$ operations in $\mathbb{K}$ by means of the *subproduct tree* technique [GG03, Chapter 10].
- If $F \in \mathbb{K}[y]$ has degree d then the remainders of F modulo all the G_i can also be computed with $\mathcal{O}(\mathsf{M}(d)\log(r))$ operations in $\mathbb{K}$. This task is usually called the *simultaneous reduction*. The inverse problem, called *Chinese remaindering*, also costs $\mathcal{O}(\mathsf{M}(d)\log(r))$[GG03, Chapter 10].

If $\mathbb{A}$ is a commutative ring with unity, and if $q \in \mathbb{A}[z]$ is a monic polynomial of degree d then an element of $\mathbb{A}[z]/(q(z))$ is represented by its coordinate vector in the usual basis $1, \dots, z^{d-1}$. Each arithmetic ring operation in $\mathbb{A}[z]/(q(z))$ reduces to $\mathcal{O}(\mathsf{M}(d))$ operations in $\mathbb{A}$.

2.1.2. *Linear Algebra.* Concerning linear algebra, we assume that, for each n, we are given a computation tree that computes the product of two $n \times n$ matrices with at most $\mathcal{O}(n^\omega)$ ring operations (*i.e.* without inversion nor division), for a fixed constant ω. We require that $2 < \omega \leq 3$ in order to use the following statement:

Lemma 5. [Sto00, particular case of Proposition 2.11] *The computation of the row reduced echelon form of a $m \times n$ matrix over $\mathbb{K}$ of rank r takes $\mathcal{O}(mnr^{\omega-2})$ operations in $\mathbb{K}$.*

Corollary 1. *If $m \geq n$ then the computation of the reduced echelon basis of the kernel of a $m \times n$ matrix over $\mathbb{K}$ takes $\mathcal{O}(mn^{\omega-1})$ operations in $\mathbb{K}$.*

Proof. Let M be such a $m \times n$ matrix. Let $\tilde{M}$ denote the $m \times n$ matrix row mirrored of M, that is $\tilde{M}_{i,j} = M_{i,n-j+1}$ for all $(i,j) \in \{1, \dots, m\} \times \{1, \dots, n\}$. By the previous proposition, the computation of the reduced echelon form of $\tilde{M}$ costs $\mathcal{O}(mn^{\omega-1})$. Then the reduced echelon basis of the kernel of M can be directly read off from the latter echelon form with $\mathcal{O}(n^2)$ operations in $\mathbb{K}$. □

2.1.3. *Extensions for our Algorithms.* In order to present a complete factorization algorithm in Section 3, we need to enlarge the computational model with irreducible factorization in $\mathbb{K}[y]$. Separately we also count the number of pth root extractions in $\mathbb{K}$. By root extraction we mean testing if an element in $\mathbb{K}$ is a pth root, and returning its casual root.

When (C) does not hold we further need to know the algebra structure of $\mathbb{K}$ over its prime field $\mathbb{F}$. Precisely we require that we can access to any component of any element in some basis over $\mathbb{F}$ for free. We also assume that any element of $\mathbb{F}$ can be sent into $\mathbb{K}$ for free. Our computational trees will thus contain operations both in $\mathbb{K}$ and $\mathbb{F}_p$. These assumptions are indeed not restrictive and cover the case when $\mathbb{K}$ is explicitly finitely generated over $\mathbb{F}_p$, that is considered in [DT81, Ste05].

For the cost analysis in positive characteristic, we will mainly focus on the case when $\mathbb{K}$ is a finite field, that $\mathbb{K} := \mathbb{F}_{p^k}$ for some $k \geq 1$. Such a finite field $\mathbb{K}$ is supposed to be given as a quotient $\mathbb{F}_p[z]/q(z)$ with q being an irreducible polynomial of degree k, so that each field operation in $\mathbb{K}$ amounts to $\mathcal{O}(\mathsf{M}(k)\log(k))$ operations in $\mathbb{F}_p$. Over finite fields, we will exclusively use computation trees over $\mathbb{F}_p$.

2.1.4. *Probabilistic Algorithms.* In order to modelize probabilistic algorithms, we extend the computation tree model with a function that takes an integer n as input, and that returns a random subset of a fixed set $\mathcal{N}$ of cardinality n, assuming that the cardinality N of $\mathcal{N}$ is at least n. The cost of this operation is assumed to be bounded by a super-additive function written $\mathsf{R}(n)$, that only depends on n. The probability distribution is supposed to be uniform in the space of subsets of $\mathcal{N}$ of cardinality n. Let us recall the following probability estimate:

Lemma 6. [Lec07a, Lemma 9] *Let $\mathcal{M}$ be a subset of $\mathcal{N}$ of cardinality M. For any $n \leq M$, the density of subsets of $\mathcal{N}$ of cardinality $2n$ having at most n elements in $\mathcal{M}$ is at least $\mathcal{E}(M,N)$, where:*

$$\begin{aligned}\mathcal{E}(M,N) &:= \frac{1}{1+\left(\frac{M}{N-M}\right)^2}, \ \textit{for } N \geq 2M\\ &:= 0, \ \textit{for } N < 2M.\end{aligned}$$

Note that $\mathcal{E}(M,N) \geq 1/2$ whenever $N \geq 2M$. Let us finally recall the classical Schwartz-Zippel lemma: the density of points in $\mathcal{N}$ that do not annihilate a given nonzero polynomial in n variables of degree d is at least $\mathcal{Z}(d,N) := 1-d/N$ [GG03, Chapter 6, Section 9].

2.2. **Deterministic Algorithm.** Propositions 1, 2 and 3 naturally lead to the following recombination algorithm:

Algorithm 1. *Deterministic recombination.*

```
Input: $F \in \mathbb{K}[x,y]$, and $\mathfrak{F}_1,\dots,\mathfrak{F}_s$ to precision $(x^{d_x+1})$.
Output: $\mu_1,\dots,\mu_r$.
1. For each $i \in \{1,\dots,s\}$, compute $\hat{\mathfrak{F}}_i$ as the quotient of $F$ by $\mathfrak{F}_i$ to precision $(x^{d_x+1})$.
2. Compute $\hat{\mathfrak{F}}_1\frac{\partial \mathfrak{F}_1}{\partial y},\dots,\hat{\mathfrak{F}}_s\frac{\partial \mathfrak{F}_s}{\partial y}$ to precision $(x^{d_x+1})$ and deduce $\mathfrak{G}_1,\dots,\mathfrak{G}_s$.
3. Compute $\mathsf{D}(\mathfrak{G}_1),\dots,\mathsf{D}(\mathfrak{G}_s)$.
4. Compute the reduced echelon solution basis $(\tilde{\mu}_1,\dots,\tilde{\mu}_t)$ of the following linear system in the unknowns $(\ell_1,\dots,\ell_s) \in \mathbb{F}^s$:
       $$\sum_{i=1}^{s} \ell_i \mathsf{D}(\mathfrak{G}_i) = 0. \qquad (4)$$
   If (C) holds then return $\tilde{\mu}_1,\dots,\tilde{\mu}_t$.
5. For all $i \in \{1,\dots,t\}$, compute $\tilde{\mathfrak{G}}_i = \sum_{j=1}^{s} \tilde{\mu}_{i,j}\mathfrak{G}_j$.
6. If $p$ divides $d_x+1$ then let $e := d_x+2$ else let $e := d_x+1$. Let $\zeta$ denote the residue class of $z$ in $\mathbb{K}[z]/(z^e-1)$. Compute $\mathsf{N}(\tilde{\mathfrak{G}}_1)(\zeta,y),\dots,\mathsf{N}(\tilde{\mathfrak{G}}_t)(\zeta,y)$.
```

7. Compute a solution basis $\nu_1, \dots, \nu_r$ of the following linear system in the unknowns $(\ell_1, \dots, \ell_t) \in \mathbb{F}^t$:
$$\sum_{i=1}^{t} \ell_i \mathsf{N}(\tilde{\mathfrak{G}}_i)(\zeta, y) = 0. \tag{5}$$
8. Compute and return the reduced echelon basis of the space $\langle \sum_{j=1}^{t} \nu_{i,j} \tilde{\mu}_j \mid i \in \{1, \dots, r\} \rangle_{\mathbb{F}}$.

Proposition 4. *Under Hypothesis (H) Algorithm 1 works correctly as specified.*

a. *If (C) holds then Algorithm 1 performs $\mathcal{O}(d_x d_y s^{\omega-1} + s\mathsf{M}(d_x)\mathsf{M}(d_y))$ operations in $\mathbb{K}$.*
b. *If (C) does not hold then Algorithm 1 performs*
$$\mathcal{O}\Big(\mathsf{M}(d_x)\big(d_y^2 s^{\omega-2} + \mathsf{M}(d_y)(s\log(s) + d_y + \log(d_x d_y)) + s d_y \log(d_x d_y)\big)\Big)$$
operations in $\mathbb{K}$ plus $\mathcal{O}(\max(e_{\mathsf{D}}, e_{\mathsf{N}}, s)s^{\omega-1})$ operations in $\mathbb{F}_p$, where e_{D} and e_{N} represent the number of equations in systems (4) and (5) respectively.

Proof. If (C) holds then Proposition 1 gives us that $(\tilde{\mu}_t, \dots, \tilde{\mu}_t) = (\mu_1, \dots, \mu_r)$. If (C) does not hold then Proposition 1 gives us that $\langle \mu_1, \dots, \mu_r \rangle_{\mathbb{F}} \subseteq \langle \tilde{\mu}_1, \dots, \tilde{\mu}_t \rangle_{\mathbb{F}}$, so that $\langle \mu_1, \dots, \mu_r \rangle_{\mathbb{F}}$ is the kernel of the restriction of $\mathsf{N}_{\mathbb{F}}$ to $\langle \tilde{\mu}_1, \dots, \tilde{\mu}_t \rangle_{\mathbb{F}}$ by Proposition 2. Since e is chosen so that $z^e - 1$ admits $e \geq d_x + 1$ distinct roots in $\bar{\mathbb{K}}$, Proposition 3 implies that $\nu_1, \dots, \nu_r$ is a basis of the kernel of the latter restriction expressed in the coordinates $\tilde{\mu}_1, \dots, \tilde{\mu}_t$. Therefore the reduced echelon basis returned in step 8 is really $\mu_1, \dots, \mu_r$.

Let us now analyze the cost of the algorithm. Testing whether (C) holds or not, and obtaining the characteristic p of $\mathbb{K}$ when (C) does not hold, takes $\mathcal{O}(d_x d_y)$ operations in $\mathbb{K}$. Steps 1 to 3 take $\mathcal{O}(s\mathsf{M}(d_x)\mathsf{M}(d_y))$ operations in $\mathbb{K}$. If (C) holds then the linear system in step 4 involves s unknowns and $\mathcal{O}(d_x d_y)$ equations. Therefore its resolution over $\mathbb{K}$ takes $\mathcal{O}(d_x d_y s^{\omega-1})$ operations in $\mathbb{K}$ by Corollary 1. We are done with part (a).

Let us now assume that (C) does not hold. The resolution of the linear system in step 4 now takes $\mathcal{O}(\max(e_{\mathsf{D}}, s)s^{\omega-1})$ operations in $\mathbb{F}$ by Corollary 1. In step 5, the computation of each $\tilde{\mathfrak{G}}_i$ can be done by means of the sub-product tree technique with $\mathcal{O}(M(d_x)\mathsf{M}(d_y)\log(s))$ operations in $\mathbb{K}$.

By Proposition 13 the cost of step 6 belongs to
$$\mathcal{O}\Big(\mathsf{M}(d_x)\big(\mathsf{M}(d_y)(d_y + \log(p)) + d_y^2 t^{\omega-2} + t d_y(\log(d_x) + \log(p))\big)\Big).$$
The resolution of the linear system in step 7 costs $\mathcal{O}(\max(e_{\mathsf{N}}, t)t^{\omega-1})$ operations in $\mathbb{F}$ by Corollary 1. In step 8 the matrix product and the reduced form computation amount to $\mathcal{O}(s^\omega)$ operations in $\mathbb{F}$ thanks to Proposition 5. Finally part (b) follows from summing the costs of all steps, and taking into account that $\log(p) \in \mathcal{O}(\log(d_x d_y))$. □

Corollary 2. *If $\mathbb{K} := \mathbb{F}_{p^k}$ then Algorithm 1 performs $\tilde{\mathcal{O}}(k d_x d_y^\omega)$ operations in $\mathbb{F}_p$.*

Proof. This follows from the previous proposition since $e_{\mathsf{D}} \in \mathcal{O}(k d_x d_y)$ and $e_{\mathsf{N}} \in \mathcal{O}(k d_x d_y)$. □

2.3. **Probabilistic Algorithm.** When (C) holds, when s is close to d_y, and when ω is close to 3, the first bottleneck of Algorithm 1 is the resolution of the linear system (4) that amounts to $\mathcal{O}(d_x d_y^\omega)$ operations in $\mathbb{K}$. When (C) does not hold, and when t is close to d_y, the second bottleneck is the evaluation of N is step 6 that also contributes to $\tilde{\mathcal{O}}(d_x d_y^\omega)$.

In this section we improve Algorithm 1 by means of probabilistic techniques. Concerning the first bottleneck we appeal to the following classical method for overdetermined linear systems. From now on we let $m := 5d_xd_y - 2d_x$ and $n := (d_x+1)d_y$. If $(u_2, \ldots, u_m) \in \mathbb{K}^{m-1}$, we write U for the following upper triangular $s \times m$ Toeplitz matrix:

$$U := \begin{pmatrix} 1 & u_2 & u_3 & \cdots & u_{m-1} & u_m \\ & 1 & u_2 & u_3 & \cdots & u_{m-1} \\ & & \ddots & \ddots & \ddots & \vdots \\ & & & 1 & \ldots & u_{m-s+1} \end{pmatrix}.$$

Similarly, if $(v_2, \ldots, v_n) \in \mathbb{K}^{n-1}$, we write V for the following upper triangular $t \times n$ Toeplitz matrix:

$$V := \begin{pmatrix} 1 & v_2 & v_3 & \cdots & v_{n-1} & v_n \\ & 1 & v_2 & v_3 & \cdots & v_{n-1} \\ & & \ddots & \ddots & \ddots & \vdots \\ & & & 1 & \ldots & v_{n-t+1} \end{pmatrix}.$$

The following lemma, derived from [KS91, Theorem 2], will provide us with a probabilistic algorithm to solve the overdetermined linear systems (4) and (5):

Lemma 7. [Lec07b, Lemma 9] *Let A be a $m \times s$ matrix over $\mathbb{K}$. For all $(u_2, \ldots, u_m)$ in $\mathbb{K}^{m-1}$ we have $\ker(A) \subseteq \ker(UA)$. In addition, there exists a nonzero polynomial $\mathcal{P} \in \mathbb{K}[z_2, \ldots, z_m]$ of total degree at most s such that the latter inclusion is an equality whenever $\mathcal{P}(u_2, \ldots, u_m) \neq 0$.*

Concerning the second bottleneck we replace the operator N by B that can be evaluated faster. The probabilistic algorithm thus proceeds as follows:

Algorithm 2. *Probabilistic recombination*

Input: $F \in \mathbb{K}[x, y]$, $\mathfrak{F}_1, \ldots, \mathfrak{F}_s$ to precision (x^{d_x+1}), and a finite subset $\mathcal{N}$ of $\mathbb{K}$ of cardinality N.
Output: $\mu_1, \ldots, \mu_r$.

Do the same computations as in Algorithm 1, but replace steps 4, 6, and 7 respectively by:

4. a. Take $(u_2, \ldots, u_m)$ at random in $\mathcal{N}^{m-1}$.
   b. For all $i \in \{1, \ldots, s\}$, compute $D_i := U\mathsf{D}(\mathfrak{G}_i)$, where $\mathsf{D}(\mathfrak{G}_i)$ is seen as a vector in $\mathbb{K}^m$.
   c. Compute the reduced echelon solution basis $\tilde{\mu}_1, \ldots, \tilde{\mu}_t$ of the following linear system in the unknowns $(\ell_1, \ldots, \ell_s) \in \mathbb{F}^s$:

$$\sum_{i=1}^{s} \ell_i D_i = 0. \tag{6}$$

   If (C) holds then return $\tilde{\mu}_1, \ldots, \tilde{\mu}_t$.
6. a. Take a finite subset of $\mathcal{N}$ at random with cardinality $2(d_x+1)$.
   b. If possible select d_x+1 points $a_0, \ldots, a_{d_x}$ from the latter set such that $c(a_i)\Delta_y(a_i) \neq 0$. If not possible then stop the execution.
   c. For all $i \in \{1, \ldots, t\}$, compute the vector A_i in $\mathbb{K}^n$ whose $jd_y + k + 1$th entry is the coefficient of y^k in $\mathsf{B}(\tilde{\mathfrak{G}}_i)(a_j, y)$, for all $k \in \{0, \ldots, d_y - 1\}$ and all $j \in \{0, \ldots, d_x\}$.
7. a. Take $(v_2, \ldots, v_n)$ at random in $\mathcal{N}^{n-1}$.

b. For all $i \in \{1, \ldots, t\}$, compute $B_i := VA_i$. Compute a solution basis $\nu_1, \ldots, \nu_r$ of the following linear system in the unknowns $(\ell_1, \ldots, \ell_t) \in \mathbb{F}^t$:

$$\sum_{i=1}^{t} \ell_i B_i = 0. \tag{7}$$

Proposition 5. *The space spanned over $\mathbb{F}$ by the output of Algorithm 2 always contains $\mu_1, \ldots, \mu_r$.*

a. *If (C) holds then Algorithm 2 takes $\mathcal{O}(s(\mathsf{M}(d_x)\mathsf{M}(d_y) + \mathsf{M}(d_x d_y)) + s^\omega)$ operations in $\mathbb{K}$, plus $\mathsf{R}(m-1)$. The probability to obtain a correct result is at least $\mathcal{Z}(s, N)$.*

b. *If (C) does not hold then Algorithm 2 takes $\mathcal{O}(s(\mathsf{M}(d_x)\mathsf{M}(d_y)\log(d_x d_y) + \mathsf{M}(d_x d_y))$ operations in $\mathbb{K}$, plus $\mathsf{R}(2d_x + m + n)$, plus $\mathcal{O}(\max(f_{\mathsf{D}}, f_{\mathsf{B}}, s)s^{\omega-1})$ operations $\mathbb{F}_p$, where f_{D} and f_{B} represent the number of equations in systems (6) and (7) respectively. The probability to obtain a correct result is at least $\mathcal{E}(2d_x d_y, N)\mathcal{Z}(2s, N)$.*

Proof. By Lemma 7, the solution space of system (6) always contains the solution space of system (4), that is $\mu_1, \ldots, \mu_r$. If (C) does not hold then, for any $i \in \{1, \ldots, r\}$, we can write $\mu_i = \ell_1 \tilde{\mu}_1 + \cdots + \ell_t \tilde{\mu}_t$ with $(\ell_1, \ldots, \ell_t) \in \mathbb{F}^t$ so that we have:

$$\sum_{j=1}^{t} \ell_j \mathsf{B}(\tilde{\mathfrak{G}}_j) = \mathsf{B}\left(\sum_{j=1}^{t} \ell_j \sum_{k=1}^{s} \tilde{\mu}_{j,k} \mathfrak{G}_k\right) = \mathsf{B}\left(\sum_{k=1}^{s} \mu_{i,k} \mathfrak{G}_k\right) = \mathsf{B}_{\mathbb{F}}(\mu_i) = 0,$$

where the last equality follows from Proposition 2. Therefore, by Lemma 7 again, the space spanned by the output of the algorithm always contains $\mu_1, \ldots, \mu_r$.

Lemma 7 provides us with a polynomial $\mathcal{P}$ such that, for any $(u_2, \ldots, u_m)$ outside the zero locus of $\mathcal{P}$, the systems (4) and (6) have the same solution sets. The probability of success thus follows from the Schwartz-Zippel lemma.

If (C) does not hold, then the probability that at most $d_x + 1$ points annihilate $c\Delta_y$ comes from Lemma 6. Then Lemma 7 provides us with a nonzero polynomial $\mathcal{Q}$ such that the solution set of (7) coincides with the solution set of the following system:

$$\sum_{i=1}^{t} \ell_i A_i = 0.$$

On the other hand, the solution set of the latter system coincides with the solution set of system (5) by Proposition 3, whence the correctness of Algorithm 2. The probability bound in part (b) follows from $\mathcal{Z}(s, N)^2 \geq \mathcal{Z}(2s, N)$.

If (C) holds then step 4 performs $\mathcal{O}(s\mathsf{M}(d_x d_y))$ operations in $\mathbb{K}$ for the computation of all the D_i, plus $\mathcal{O}(s^\omega)$ operations in $\mathbb{K}$ for the resolution of system (6), by Corollary 1. We are done with part (a).

Concerning part (b), the evaluation of $c\Delta_y$ at $2(d_x + 1)$ in step 6 can be done with $\mathcal{O}(d_y \mathsf{M}(d_x)\log(d_x) + d_x \mathsf{M}(d_y)\log(d_y))$. Then $F(a_0, y), \ldots, F(a_{d_x}, y)$, and each tuple $(\tilde{\mathfrak{G}}_i(a_0, y), \ldots, \tilde{\mathfrak{G}}_i(a_{d_x}, y))$ can be computed with $\mathcal{O}(d_y \mathsf{M}(d_x)\log(d_x))$ operations in $\mathbb{K}$. Each $\mathsf{B}(\tilde{\mathfrak{G}}_i)(a_j, y)$ is then obtained as the remainder of $\tilde{\mathfrak{G}}_i(a_j, y)^p - (F(a_j, y)')^{p-1}\tilde{\mathfrak{G}}_i(a_j, y)^p$ divided by $F(a_j, y)$ with $\mathcal{O}(\mathsf{M}(d_y)\log(p))$ operations in $\mathbb{K}$. Therefore the total cost of step 6 amounts to $\mathcal{O}(t(d_y \mathsf{M}(d_x)\log(d_x) + d_x \mathsf{M}(d_y)\log(p)))$ operations in $\mathbb{K}$.

In step 7 the computation of all the B_i takes $\mathcal{O}(t\mathsf{M}(d_x d_y))$ operations in $\mathbb{K}$, and the resolution of the linear system costs $\mathcal{O}(\max(f_{\mathsf{B}}, t)t^{\omega-1})$ operations in $\mathbb{F}$, by Corollary 1. The total cost follows from summing the costs of each step and taking into account that $p \in \mathcal{O}(d_x d_y)$. □

Corollary 3. *If $\mathbb{K} := \mathbb{F}_{p^k}$ then Algorithm 2 performs $\tilde{\mathcal{O}}(k(d_x d_y^2 + d_y^\omega))$ operations in $\mathbb{F}_p$, plus $\mathsf{R}(\mathcal{O}(d_x d_y))$. The probability to obtain a correct answer is at least $\mathcal{E}(2d_x d_y, N)\mathcal{Z}(2d_y, N)$.*

Proof. It suffices to set $f_{\mathsf{D}} \in \mathcal{O}(ks)$ and $f_{\mathsf{B}} \in \mathcal{O}(ks)$ in Proposition 5. □

2.4. **Testing the Recombination and Recovering the Factors.** In this subsection we describe how to test whether the output of Algorithm 2 is correct or not. If correct then we also recover the irreducible factors of F. If not correct then the execution is stopped. Let $\nu_1, \ldots, \nu_t$ now represent the output of Algorithm 2. From Proposition 5 we know that $\langle \mu_1, \ldots, \mu_r \rangle_{\mathbb{F}} \subseteq \langle \nu_1, \ldots, \nu_t \rangle_{\mathbb{F}}$.

Algorithm 3. *Recovering the irreducible factors.*

Input: $F \in \mathbb{K}[x, y]$, $\mathfrak{F}_1, \ldots, \mathfrak{F}_s$ to precision (x^{d_x+1}), and a reduced echelon family $\nu_1, \ldots, \nu_t$ in $\mathbb{F}^s$ such that $\langle \mu_1, \ldots, \mu_r \rangle_{\mathbb{F}} \subseteq \langle \nu_1, \ldots, \nu_t \rangle_{\mathbb{F}}$.
Output: $F_1, \ldots, F_r$.

```
1. If $t = 1$ then return $F$.
2. If the entries of the $\nu_i$ are not all in $\{0, 1\}$ then stop the execution.
3. If the supports of $\nu_1, \ldots, \nu_t$ do not form a partition of $\{1, \ldots, s\}$ of size $t$
   then stop the execution.
4. For all $i \in \{1, \ldots, t\}$, compute $\tilde{F}_i := \left\lceil c \prod_{j=1}^s \mathfrak{F}_j^{\nu_{i,j}} \right\rceil^{d_x+1}$.
5. For all $i \in \{1, \ldots, t\}$, compute $\check{F}_i$ as the primitive part of $\tilde{F}_i$ in $y$. Normalize
   $\check{F}_i$ in $\mathbb{K}[x][y]$ so that its leading coefficient becomes monic.
6. If $\prod_{i=1}^t \check{F}_j = F$ then return $\check{F}_1, \ldots, \check{F}_t$. Otherwise stop the execution.
```

Proposition 6. *Algorithm 3 returns a correct answer if, and only if, $t = r$. Otherwise the execution is stopped. The algorithm performs $\mathcal{O}(\mathsf{M}(d_x)\mathsf{M}(d_y)\log(d_x d_y) + s^2)$ arithmetic operations in $\mathbb{K}$.*

Proof. Let us first examine the case when $t = r$. If $r = 1$ then the output is clearly correct. If $r \geq 2$ then, for all $i \in \{1, \ldots, r\}$, we have that $\tilde{F}_i = c/c_i F_i$ because $\deg_x(c/c_i) + \deg_x(F_i) \leq \deg_x(F_1) + \cdots + \deg_x(F_r) = d_x$. Finally $\check{F}_i = F_i$ holds and the algorithm returns a correct answer.

Let us now show that the algorithm always return a correct result. If $t = 1$ then the algorithm exits at the first step with $t = r$. If the algorithm exits in step 2 or 3 then we necessary have $t > r$. If the algorithm reaches the last step then F admits at least $t \geq r$ irreducible factors, whence $t = r$.

Steps 1 to 3 take $\mathcal{O}(st)$ operations in $\mathbb{K}$. Letting $d_i := \deg_y(\tilde{F}_i)$, each $\tilde{F}_i$ can be computed by the sub-product tree technique with $\mathcal{O}(\mathsf{M}(d_x)\mathsf{M}(d_i)\log(d_i))$ operations, so that the total cost of step 4 amounts to $\mathcal{O}(\mathsf{M}(d_x)\mathsf{M}(d_y)\log(d_y))$. In step 5 each primitive part computation amounts to $\mathcal{O}(d_i \mathsf{M}(d_x)\log(d_x))$ operations, whence a total cost in $\mathcal{O}(d_y \mathsf{M}(d_x)\log(d_x))$ for this step. In step 6, one has to check whether $\sum_{i=1}^t \deg_x(\check{F}_i) = d_x$ holds or not before computing the product. This way the cost of step 6 drops to another $\mathcal{O}(\mathsf{M}(d_x)\mathsf{M}(d_y)\log(d_y))$. □

3. The Whole Factorization Algorithm

So far we have dealt with the factorization of normalized polynomials, namely under Hypothesis (H). In this section we explain how to handle the general case by means of successive reductions to (H). For the sake of simplicity, we use fast multiplication everywhere, namely $\mathsf{M}(d) \in \tilde{\mathcal{O}}(d)$.

3.1. **Factorization of Separable Polynomials.** Recall that a polynomial F is said to be separable in y if its discriminant Δ_y, is nonzero. The reduction of the factorization of separable polynomials to normalized polynomials simply consists in searching for a suitable shift of the variable x. The following algorithm summarizes the main steps:

Algorithm 4. *Factorization of separable polynomials.*

```
Input: $F \in \mathbb{K}[x,y]$ primitive and separable in $y$.
Output: the irreducible factors $F_1, \ldots, F_r$ of $F$.
1. Find $b \in \mathbb{K}$ such that $F(b,y)$ is separable of degree $d_y$. Replace $F$ by $F(x+b,y)$
2. Compute the irreducible decomposition of $F(0,y)$.
3. Compute $\mathfrak{F}_1, \ldots, \mathfrak{F}_s$ to precision $(x^{d_x+1})$ by means of Hensel lifting.
4. Recombine the lifted factors by means of the algorithms presented in the previous section in order to obtain the irreducible factors $F_1, \ldots, F_r$ of $F$ over $\mathbb{K}$.
5. For all $i \in \{1, \ldots, r\}$ replace $F_i$ by $F_i(x-b,y)$.
```

With deterministic subroutines we obtain the following estimates:

Proposition 7. *Assume that F is primitive and separable in y, and that $\mathbb{K}$ has cardinality at least $2d_xd_y+1$. Algorithm 4 is correct and performs one irreducible factorization in $\mathbb{K}[y]$ in degree d_y, plus*

a. *if (C) holds, $\mathcal{O}(d_xd_y^\omega)$ operations in $\mathbb{K}$;*
b. *if $\mathbb{K} := \mathbb{F}_{p^k}$, $\tilde{\mathcal{O}}(kd_xd_y^\omega)$ operations in $\mathbb{F}_p$.*

Proof. Since $\deg(c\Delta_y) \leq 2d_xd_y$, the assumption on the cardinality of $\mathbb{K}$ ensures us to find a suitable b. This search can be done with $\tilde{\mathcal{O}}(d_xd_y^2)$ operations by means of fast multi-point evaluation. Then the shift of the variable x can be done in softly optimal time via evaluation/interpolation.

In steps 2 to 4 Hypothesis (H) holds. The computation of $\mathfrak{F}_1, \ldots, \mathfrak{F}_s$ to precision (x^{d_x+1}) from the irreducible factors of $F(0,y)$ costs $\tilde{\mathcal{O}}(d_xd_y)$ operations in $\mathbb{K}$ by [GG03, Theorem 15.18] (see also [BLS+04] for implementation details). Then part (a) follows from Propositions 4 and 6. Then part (b) is completed thanks to Corollary 2. □

As for the probabilistic version, we call Algorithm 2 in step 4. For doing so we need to provide the algorithm with a subset $\mathcal{N}$ of $\mathbb{K}$ of cardinality N, used for random element generation. With these modifications we obtain:

Proposition 8. *Assume that F is primitive and separable in y, and that $\mathbb{K}$ has cardinality at least $2d_xd_y+1$. Algorithm 4 either stop prematurely or returns a correct answer. It takes one irreducible factorization in $\mathbb{K}[y]$ in degree d_y, plus*

a. *if (C) holds, $\tilde{\mathcal{O}}(d_xd_y^2+d_y^\omega)$ operations in $\mathbb{K}$ and $\mathsf{R}(\mathcal{O}(d_xd_y))$, with a probability of success at least $\mathcal{Z}(d_y,N)$;*
b. *if $\mathbb{K} := \mathbb{F}_{p^k}$, $\tilde{\mathcal{O}}(k(d_xd_y^2+d_y^\omega))$ operations in $\mathbb{F}_p$ and $\mathsf{R}(\mathcal{O}(d_xd_y))$, with a probability of success at least $\mathcal{E}(2d_xdy,N)\mathcal{Z}(2d_y,N)$.*

Proof. The proof is the same as for the deterministic case, except that we appeal to Propositions 5 and Corollary 3. □

3.2. **The General Case.** In order to raise the separability assumption on F, we use the separable factorization algorithm designed in [Lec07a]. Let

$$\mathcal{B} := \{1, p, p^2, p^3, \ldots\}$$

be the set of the powers of p. If F is primitive in y, and if $p > 0$, then the *separable decomposition* of F in y is defined to be the set

$$\{(G_1, q_1, m_1), \ldots, (G_s, q_s, m_s)\} \subseteq (\mathbb{K}[x,y] \setminus \mathbb{K}[x]) \times \mathcal{B} \times \mathbb{N}$$

satisfying the following properties:

(S_1) $F(y) = \prod_{i=1}^{s} G_i(y^{q_i})^{m_i}$;
(S_2) for all $i \neq j$ in $\{1, \ldots, s\}$, $G_i(y^{q_i})$ and $G_j(y^{q_j})$ are coprime;
(S_3) for all $i \in \{1, \ldots, s\}$, p does not divide m_i;
(S_4) for all $i \in \{1, \ldots, s\}$, G_i is separable, primitive and of positive degree in y;
(S_5) for all $i \neq j$ in $\{1, \ldots, s\}$, $(q_i, m_i) \neq (q_j, m_j)$.

If $p = 0$ then the separable decomposition of F is naturally defined to be the set generated by all the triples $(G, 1, m)$ such that G is a proper squarefree factor of F with multiplicity $m \geq 1$. For the existence, the uniqueness, and a short history of the separable decomposition we refer the reader to [Lec07a].

Under the assumption that $\mathbb{K}$ has sufficiently many elements we finally obtain the following top level factorization algorithm:

Algorithm 5. *Top level factorization algorithm.*

```
Input: $F \in \mathbb{K}[x,y]$.
Output: the irreducible factors $F_1, \ldots, F_r$ of $F$ together with their respective
    multiplicities $e_1, \ldots, e_r$.
1. Compute the content and the primitive part of $F$ in $y$. Replace $F$ by its
   primitive part and initialize the list $L$ with the irreducible decomposition of
   the content.
2. Compute the separable decomposition $(G_1, q_1, m_1), \ldots, (G_s, q_s, m_s)$ of $F$.
3. Compute the irreducible decomposition of $G_1, \ldots, G_s$.
4. For all $i \in \{1, \ldots, s\}$, and for all irreducible factor $E$ of $G_i$ do:
      If $p = 0$ then append $(E, m_i)$ to the list $L$,
      else compute $(H, h) \in \mathbb{K}[x,y] \times \mathcal{B}$ such that $H^h(y^{q_i/h}) = E(y^{q_i})$, $h \leq q_i$
      and $h$ is as large as possible, and append $(H(y^{q_i/h}), hm_i)$ to $L$.
5. Return $L$.
```

By means of deterministic subroutines we obtain:

Proposition 9. *Assume that $\mathbb{K}$ has cardinality at least $2d_xd_y + d_x + 1$. Then Algorithm 5 performs irreducible factorizations of univariate polynomials over $\mathbb{K}$ whose degree sum is at most $d_x + d_y$, plus*

a. *if (C) holds, $\mathcal{O}(d_xd_y^{\omega})$ arithmetic operations in $\mathbb{K}$;*
b. *if $\mathbb{K} := \mathbb{F}_{p^k}$, $\tilde{\mathcal{O}}(kd_xd_y^{\omega})$ arithmetic operations in $\mathbb{F}_p$.*

Proof. In Algorithm 5, the computation of the primitive part can be done in softly optimal time. Then step 2 takes $\tilde{\mathcal{O}}(d_xd_y^2)$ by [Lec07a, Proposition 8] thanks to the hypothesis on the cardinality of $\mathbb{K}$. Step 3 can be done by means of Algorithm 4: Proposition 7(a) yields a total cost in $\mathcal{O}(d_xd_y^{\omega})$ operations in $\mathbb{K}$. We are done with part (a).

As for part (b), computing each pair (H, h) costs $\mathcal{O}(\deg_x(E)\deg_y(E))$ extractions of pth roots in $\mathbb{F}_{p^k}$. One pth root extraction in $\mathbb{F}_{p^k}$ requires $\mathcal{O}((k-1)\log(p))$ arithmetic operations in $\mathbb{F}_{p^k}$. The conclusion thus follows from Proposition 7(b). □

In order to use probabilistic subroutines in Algorithm 5 we add an extra input $\mathcal{N}$ that is a subset of $\mathbb{K}$ of cardinality N used for the random choices.

Proposition 10. *Assume that $\mathbb{K}$ has cardinality at least $2d_xd_y + 1$. Then Algorithm 5 either returns a correct answer or stops prematurely. It performs irreducible* *factorizations of univariate polynomials over $\mathbb{K}$ whose degree sum is at most d_x+d_y, plus*

a. *if (C) holds, $\mathcal{O}(d_x d_y^2+d_y^\omega)$ arithmetic operations in $\mathbb{K}$, plus $\mathsf{R}(\mathcal{O}(d_x d_y))$ with a probability of success at least $\mathcal{E}(2d_x d_y, N)\mathcal{Z}(d_y, N)$;*

b. *if $\mathbb{K} := \mathbb{F}_{p^k}$, $\tilde{\mathcal{O}}(k(d_x d_y^2+d_y^\omega))$ arithmetic operations in $\mathbb{F}_p$, plus $\mathsf{R}(\mathcal{O}(d_x d_y))$ for random set generation in $\mathbb{K}$, with a probability of success at least*

$$\mathcal{E}(4d_x dy, N)\mathcal{Z}(2d_y, N).$$

Proof. Step 2 of Algorithm 5 takes $\tilde{\mathcal{O}}(d_x d_y)$ operations in $\mathbb{K}$ plus $\mathsf{R}(\mathcal{O}(d_x d_y))$, with a probability of success at least $\mathcal{E}(2d_x d_y, N)$ by [Lec07a, Proposition 9]. The proof thus follows as for the previous proposition thanks to Proposition 8. In part (b) the probability bound makes use of $\mathcal{E}(2d_x dy, N)^2 \geq \mathcal{E}(4d_x dy, N)$ [Lec07a, Lemma 12]. □

Proof of Theorem 1. Without loss of generality we can assume that $d_y \leq d_x$ so that we have $\tilde{\mathcal{O}}(d_x d_y^\omega) \subseteq \tilde{\mathcal{O}}((d_x d_y)^{(\omega+1)/2})$. The conclusion thus follows from Proposition 9. □

Proof of Theorem 2. We also reduce to the case when $d_y \leq d_x$ so that we have $\tilde{\mathcal{O}}(d_x d_y^2 + d_y^\omega) \subseteq \tilde{\mathcal{O}}((d_x d_y)^{1.5})$. Thanks to the hypothesis on the cardinality of $\mathbb{K}$ we can take $\mathcal{N}$ with cardinality $N = 10 d_x d_y$ so that $\mathcal{E}(4d_x dy, N)\mathcal{Z}(2d_y, N) > 1/2$ holds. The conclusion follows from Proposition 10. □

Conclusion

Although the factorization algorithms presented in this paper have good worst case complexity bounds their implementations require some care to make them very efficient in practice. The first bottleneck of the algorithm underlying Theorem 2 is the computation of the polynomials $\mathfrak{G}_i$. This bottleneck can be avoided by adapting the heuristic presented in [BLS$^+$04, Lec06, Lec07b]. Then the next practical bottleneck really becomes the Hensel lifting. Although this lifting can be done in softly optimal time, one interesting speedup consists in solving the recombination problem progressively at each stage of the lifting in order to decrease the number of the lifted factors and stop the lifting as soon as possible. Finally, in a good implementation, the worst case precision bound for the lifting is hardly never attained. These heuristics and others have been implemented by Steel in Magma [Mag] on the top of the algorithms designed in [Hoe02, BHKS04, Ste05]. Our new algorithms are not intended to compete in general with Steel's implementation. Instead they sensibly improve the performances in very particular cases. Such cases are very difficult to build. One example is provided by [BHKS04, Remark 5.5].

Appendix A. Berlekamp's and Niederreiter's Equations

This appendix gathers some classical properties of Berlekamp's and Niederreiter's operators used in Section 1, and contains a cost analysis of the evaluation of Niederreiter's operator used in Section 2. From now on $\mathbb{K}$ denotes a field of characteristic $p > 0$, whose prime field is written $\mathbb{F}$.

A.1. Classical Properties.

Let f be a monic separable polynomial of degree d in $\mathbb{K}[x]$, and let $g \in \mathbb{K}[x]$ be of degree at most $d-1$. Let $\mathbb{E}$ be a field extension of $\mathbb{K}$ containing all the roots $\varphi_1, \ldots, \varphi_d$ of f. The kth derivative of $h \in \mathbb{K}[x]$ is written $h^{(k)}$.

Proposition 11. *The following assertions are equivalent:*

a. *$g(\varphi_i)/f'(\varphi_i) \in \mathbb{F}$, for all $i \in \{1, \ldots, d\}$.*

b. $g^p - (f')^{p-1} g \in (f)$ *(Berlekamp's operator).*
c. $g^p + (f^{p-1} g)^{(p-1)} = 0$ *(Niederreiter's operator).*

Proof. In short let ρ_i represent $g(\varphi_i)/f'(\varphi_i)$ for each $i \in \{1, \dots, d\}$, so that the partial fraction decomposition of g/f over $\mathbb{E}$ reads as:

$$\frac{g}{f} = \sum_{i=1}^{d} \frac{\rho_i}{x - \varphi_i}.$$

For any $i \in \{1, \dots, d\}$ we have that $(g^p - (f')^{p-1} g)(\varphi_i) = (f')^p(\varphi_i)(\rho_i^p - \rho_i)$, which gives the equivalence between parts (a) and (b). The equivalence between parts (a) and (c) follows from the following calculation:

$$\begin{aligned} g^p + (f^{p-1} g)^{(p-1)} &= g^p + \left(f^p \frac{g}{f}\right)^{(p-1)} = g^p + f^p \left(\frac{g}{f}\right)^{(p-1)} \\ &= f^p \left(\left(\frac{g}{f}\right)^p + \left(\frac{g}{f}\right)^{(p-1)}\right) \\ &= f^p \left(\sum_{i=1}^{d} \frac{\rho_i^p}{(x - \phi_i)^p} + \sum_{i=1}^{d} \frac{-\rho_i}{(x - \phi_i)^p}\right), \end{aligned} \tag{8}$$

where the latter equality makes use of Wilson's theorem (namely, p divides $(p-1)! + 1$). □

A.2. **Complexity.** Let $\mathbb{A}$ be any commutative ring with unity of characteristic p, and let f be a polynomial in $\mathbb{A}[x]$ of degree d. We are to study the cost for computing $g^p + (f^{p-1} g)^{(p-1)}$ for a given $g \in \mathbb{A}[x]_{d-1}$. In the latter expression, the most difficult part is the computation of $(f^{p-1} g)^{(p-1)}$. We thus focus on the construction of the $d \times d$ matrix N of the following map expressed in the usual monomial basis:

$$\begin{aligned} \mathbb{A}[x]_{d-1} &\to \mathbb{A}[x^p]_{d-1} \\ g &\mapsto (f^{p-1} g)^{(p-1)}. \end{aligned}$$

Recall that $\mathbb{A}[x^p]_{d-1}$ denotes the space of polynomials in x^p of degree at most $d-1$ in x^p.

Proposition 12. *If f is monic then N can be computed with $\mathcal{O}(\mathsf{M}(d)(d + \log(p)))$ operations in $\mathbb{A}$.*

Proof. We write $\operatorname{coeff}(A, x^i)$ for the coefficient of x^i in $A \in \mathbb{A}[x]$. We aim to compute $-\operatorname{coeff}(f^{p-1} g, x^{ip+p-1})$ for all $i \in \{0, \dots, d-1\}$. Let $\operatorname{rev}(n, .)$ denote the reversal endomorphism of $\mathbb{A}[x]_n$ defined by $\operatorname{rev}(n, A) = \sum_{k=0}^{n} a_{n-k} x^k$, for all $A := \sum_{k=0}^{n} a_k \in \mathbb{A}[x]_n$. In fact we will compute

$$-\operatorname{coeff}(\operatorname{rev}(dp - 1, f^{p-1} g), x^{ip}), \text{ for all } i \in \{0, \dots, d-1\}.$$

Since f is monic of degree d, the constant term of $\operatorname{rev}(d, f)$ is 1. Hence we can define $\sum_{i \geq 0} u_i x^i$ as the power series expansion of $\operatorname{rev}(d-1, g)/\operatorname{rev}(d, f)$, so that we have:

$$\begin{aligned} \operatorname{rev}(dp - 1, f^{p-1} g) &= \operatorname{rev}(d, f)^{p-1} \operatorname{rev}(d-1, g) \\ &= \operatorname{rev}(d, f)^p \operatorname{rev}(d-1, g)/\operatorname{rev}(d, f) \\ &= \operatorname{rev}(d, f)^p \sum_{i \geq 0} u_i x^i. \end{aligned}$$

Therefore we are led to compute $u_0, u_p, \ldots, u_{(d-1)p}$. The computation of $\operatorname{rev}(d, f)^p$ takes $\mathcal{O}(d \log(p))$ operations in $\mathbb{A}$, and then the product

$$\operatorname{rev}(d, f)^p \sum_{i=0}^{d-1} u_{ip} x^{ip}$$

costs $\mathcal{O}(\mathsf{M}(d))$. In order to compute $u_0, u_p, \ldots, u_{(d-1)p}$, we are going to use the fact that $(u_i)_{i \in \mathbb{N}}$ satisfies the following linear recurrence:

$$u_i := -(f_{d-1} u_{i-1} + \cdots + f_0 u_{i-d}), \text{ for } i \geq d,$$

where $f_j := \operatorname{coeff}(f, x^j)$ (recall that $f_d = 1$). Of course the initial values $u_0, \ldots, u_{d-1}$ can be computed from the power series expansion of $\operatorname{rev}(d-1, g)/\operatorname{rev}(d, f)$ to precision (x^d) with $\mathcal{O}(\mathsf{M}(d))$ operations in $\mathbb{A}$.

For all $i \geq 0$, if $a_{d-1}x^{d-1} + \cdots + a_1 x + a_0$ represents the remainder $\operatorname{rem}(x^i, f)$ in the division of x^i by f, then it is classical that $u_i = a_0 u_0 + \cdots + a_{d-1} u_{d-1}$. The idea of using the latter formula to efficiently compute u_i is due to Fiduccia [Fid85]. This formula expresses the fact that linear recurrence sequence extension is transposed to the remainder operation (see [BLS03, Section 5] for details). Let B be the $d \times d$ matrix whose ith column is $\operatorname{rem}(x^{p(i-1)}, f)$ expressed in the usual basis $1, x, \ldots, x^{d-1}$. This is nothing else but the Petr-Berlekamp matrix arising in univariate polynomial factorization over finite fields. The construction of B takes $\mathcal{O}(\mathsf{M}(d)(d+\log(p)))$ operations in $\mathbb{A}$. Finally we have shown that N can be factored into $N = H_2 B^t H_1$, where B^t is the transpose of B, and H_1, H_2 are the Hankel matrices of the following maps:

$$\begin{aligned} H_1 : & \mathbb{A}[x]_{d-1} \to \mathbb{A}^d \\ & g \mapsto (u_0, \ldots, u_{d-1}), \\ H_2 : & \mathbb{A}^d \to \mathbb{A}[x^p]_{d-1} \\ & (u_0, \ldots, u_{(d-1)p}) \mapsto -\operatorname{rev}\left(dp-1, \left\lceil \operatorname{rev}(d, f)^p \sum_{i=0}^{d-1} u_{ip} x^{ip} \right\rceil^{dp}\right). \end{aligned}$$

Recall that $\lceil A \rceil^l$ represents the projection of the series $A := \sum_{i \geq 0} a_i x^i \in \mathbb{K}[[x]]$ to the polynomial $\sum_{i=0}^{l-1} a_i x^i \in \mathbb{K}[x]_{l-1}$. Since H_1 is symmetric, $(B^t H_1)^t = H_1 B$ can be computed with $\mathcal{O}(d\mathsf{M}(d))$ operations in $\mathbb{A}$. The second product $H_2(B^t H_1)$ also costs $\mathcal{O}(d\mathsf{M}(d))$. □

The above decomposition $N = H_2 B^t H_1$ was already stated in [Nie94, Theorem 1] (in the more general setting of the Hasse-Teichmüller derivative). Unfortunately the cost analysis made in [Nie94, Theorem 2] was pessimistic because it did not take into account that left and right products by Hankel matrices could be done in softly optimal time.

From now on we assume that $\mathbb{A} := \mathbb{K}[z]/(q(z))$, where q is a non-constant monic separable polynomial in $\mathbb{K}[z]$ of degree n. Let $g_1, \ldots, g_t$ be a sequence of polynomials in $\mathbb{A}[x]_{d-1}$ with $1 \leq t \leq d$. We are interested in evaluating Niederreiter's operator simultaneously at each g_i.

Proposition 13. *The computation of $g_1^p + (f^{p-1}g_1)^{(p-1)}, \ldots, g_t^p + (f^{p-1}g_t)^{(p-1)}$ takes*

$$\mathcal{O}\Big(\mathsf{M}(n)\big(\mathsf{M}(d)(d+\log(p)) + d^2 t^{\omega-2} + td(\log(n)+\log(p))\big)\Big)$$

operations in $\mathbb{K}$.

Proof. The computation of $g_1^p, \ldots, g_t^p$ can be done with $\mathcal{O}(td\log(p))$ ring operations in $\mathbb{A}$, which amounts to $\mathcal{O}(td\mathsf{M}(n)\log(p))$ operations in $\mathbb{K}$. We can now focus on the computation of $(f^{p-1}g_1)^{(p-1)}, \ldots, (f^{p-1}g_t)^{(p-1)}$. If f is monic then this computation can be achieved by means of the matrix N of Proposition 12 that can be built with $\mathcal{O}(\mathsf{M}(d)(d+\log(p)))$ operations in $\mathbb{A}$: it suffices to calculate the product of N with the $d \times t$ matrix whose columns are the coefficients of the g_i. This product takes $\mathcal{O}(d^2t^{\omega-2})$ operations in $\mathbb{A}$. We are done with the monic case.

If f is not monic then we decompose $\mathbb{A}$ into $\mathbb{A}_1 \times \cdots \times \mathbb{A}_r$ so that f can be made monic in each $\mathbb{A}_i$. Each $\mathbb{A}_i$ is of the form $\mathbb{A}_i := \mathbb{K}[z]/(q_i(z))$, where q_i will be a monic polynomial in $\mathbb{K}[z]$ if degree $n_i \geq 1$. We perform the computations in each $\mathbb{A}_i$ and recover the result by means of Chinese remaindering. This decomposition is obtained as follows. For all $i \in \{0, \ldots, d\}$, let f_i represent the canonical preimage in $\mathbb{K}[z]$ of the coefficient of x^i in f. We construct $\mathbb{A}_1 := \mathbb{K}[z]/(q_1(z))$ so that the projection of f into $\mathbb{A}_1[x]$ has an invertible leading coefficient. More precisely, q_1 is obtained as $q_1 := q/\gcd(f_d, q)$. Since q is separable, q/q_1 is prime with q_1. It follows that f_d is invertible in $\mathbb{A}_1$. Let k_1 be the largest integer such that f_{k_1} is nonzero modulo q/q_1. Of course we have $k_1 \leq d-1$, and k_1 can be found with $(d-k_1)\mathsf{M}(n)$ operations in $\mathbb{K}$. From f_{k_1} and q/q_1 we can compute q_2 such that f_{k_1} is invertible in $\mathbb{A}_2 = \mathbb{K}[z]/(q_2(z))$ and reduces to zero modulo $q/(q_1q_2)$. By iterating this process we obtain the claimed decomposition with $\mathcal{O}(d\mathsf{M}(n)\log(n))$ operations in $\mathbb{K}$.

By [GG03, Corollary 10.7] any element of $\mathbb{A}$ can be sent to $\mathbb{A}_1 \times \cdots \times \mathbb{A}_r$ with $\mathcal{O}(\mathsf{M}(n)\log(n))$ operations in $\mathbb{K}$. Therefore all the projections of f and the g_i into $\mathbb{A}_1[x] \times \cdots \times \mathbb{A}_r[x]$ amounts to $\mathcal{O}(td\mathsf{M}(n)\log(n))$.

Now let $i \in \{1, \ldots, r\}$ and let us analyze the computation of

$$(f^{p-1}g_1)^{(p-1)}, \ldots, (f^{p-1}g_s)^{(p-1)}$$

in $\mathbb{A}_i[x]$. For each $j \in \{1, \ldots, t\}$ we introduce π_j and ρ_j for the quotient and the remainder of g_j divided by f in $\mathbb{A}_i$, so that $g_j = \pi_j f + \sigma_j$ holds in $\mathbb{A}_i$. By construction the divisions are well defined since the leading coefficient of f is invertible in $\mathbb{A}_i$. The inversion of the leading coefficient of f in $\mathbb{A}_i$ takes $\mathcal{O}(\mathsf{M}(n_i)\log(n_i))$ operations in $\mathbb{K}$. Then the computation of all the π_j and σ_j amounts to $\mathcal{O}(t\mathsf{M}(d))$ ring operations in $\mathbb{A}_i$. For each j we thus have to compute

$$(f^{p-1}g_j)^{(p-1)} = (f^p\pi_j)^{(p-1)} + (f^{p-1}\sigma_j)^{(p-1)} = f^p\pi_j^{(p-1)} + (f^{p-1}\sigma_j)^{(p-1)}.$$

The computation of all the $(f^{p-1}\sigma_j)^{(p-1)}$ can be done with $\mathcal{O}(d^2t^{\omega-2} + \mathsf{M}(d)(d + \log(p)))$ ring operations in $\mathbb{A}_i$. Computing all the $f^p\pi_j^{(p-1)}$ costs $\mathcal{O}(d\log(p) + t\mathsf{M}(d))$ operations in $\mathbb{A}_i$. The computation of $(f^{p-1}g_1)^{(p-1)}, \ldots, (f^{p-1}g_s)^{(p-1)}$ thus amounts to $\mathcal{O}(d^2t^{\omega-2}+\mathsf{M}(d)(d+\log(p)))$ ring operations in $\mathbb{A}_i$ plus $\mathcal{O}(\mathsf{M}(n_i)\log(n_i))$ operations in $\mathbb{K}$.

The sum of these costs over $i \in \{1, \ldots, r\}$ leads to $\mathcal{O}(\mathsf{M}(n)(\mathsf{M}(d)(d + \log(p)) + d^2t^{\omega-2} + \log(n)))$ thanks to the super-additivity of M. Finally, all the $(f^{p-1}g_i)^{(p-1)}$ can be lifted into $\mathbb{A}[x]$ by Chinese remaindering with $\mathcal{O}(td\mathsf{M}(n)\log(n))$ operations in $\mathbb{K}$ by [GG03, Corollary 10.23]. □

References

[BCS97] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory*. Springer-Verlag, 1997.

[BHKS04] K. Belabas, M. van Hoeij, J. Klüners, and A. Steel. Factoring polynomials over global fields. Manuscript available at http://arxiv.org/abs/math.NT/0409510, September 2004.

[BLS03] A. Bostan, G. Lecerf, and É. Schost. Tellegen's principle into practice. In *ISSAC '03: Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 37–44. ACM Press, 2003.

[BLS$^+$04] A. Bostan, G. Lecerf, B. Salvy, É. Schost, and B. Wiebelt. Complexity issues in bivariate polynomial factorization. In *ISSAC '04: Proceedings of the 2004 international symposium on Symbolic and algebraic computation*, pages 42–49. ACM Press, 2004.

[CG05] G. Chèze and A. Galligo. Four lectures on polynomial absolute factorization. In A. Dickenstein and I. Z. Emiris, editors, *Solving polynomial equations: foundations, algorithms, and applica tions*, volume 14 of *Algorithms Comput. Math.*, pages 339–392. Springer-Verlag, 2005.

[DT81] J. H. Davenport and B. M. Trager. Factorization over finitely generated fields. In *SYMSAC '81: Proceedings of the fourth ACM symposium on Symbolic and algebraic computation*, pages 200–205. ACM Press, 1981.

[Fid85] C. M. Fiduccia. An efficient formula for linear recurrences. *SIAM J. Comput.*, 14(1):106–112, 1985.

[Gao03] S. Gao. Factoring multivariate polynomials via partial differential equations. *Math. Comp.*, 72(242):801–822, 2003.

[Gat84] J. von zur Gathen. Hensel and Newton methods in valuation rings. *Math. Comp.*, 42(166):637–661, 1984.

[GG03] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, second edition, 2003.

[GL02] S. Gao and A. G. B. Lauder. Hensel lifting and bivariate polynomial factorisation over finite fields. *Math. Comp.*, 71(240):1663–1676, 2002.

[Hoe02] M. van Hoeij. Factoring polynomials and the knapsack problem. *J. Number Theory*, 95(2):167–189, 2002.

[Kal82a] E. Kaltofen. Polynomial factorization. In B. Buchberger, G. Collins, and R. Loos, editors, *Computer algebra*, pages 95–113. Springer-Verlag, 1982.

[Kal82b] E. Kaltofen. A polynomial reduction from multivariate to bivariate integral polynomial factorization. In *Proceedings of the 14th Symposium on Theory of Computing*, pages 261–266. ACM, 1982.

[Kal82c] E. Kaltofen. A polynomial-time reduction from bivariate to univariate integral polynomial factorization. In *Proceedings of the 23rd Symposium on Foundations of Computer Science*, pages 57–64. IEEE, 1982.

[Kal85] E. Kaltofen. Sparse Hensel lifting. In *EUROCAL'85, Vol. 2 (Linz, 1985)*, volume 204 of *LNCS*, pages 4–17. Springer-Verlag, 1985.

[Kal90] E. Kaltofen. Polynomial factorization 1982–1986. In *Computers in mathematics (Stanford, CA, 1986)*, volume 125 of *Lecture Notes in Pure and Appl. Math.*, pages 285–309. Dekker, 1990.

[Kal92] E. Kaltofen. Polynomial factorization 1987–1991. In *LATIN '92 (São Paulo, 1992)*, volume 583 of *Lecture Notes in Comput. Sci.*, pages 294–313. Springer-Verlag, 1992.

[Kal95] E. Kaltofen. Effective Noether irreducibility forms and applications. *J. Comput. System Sci.*, 50(2):274–295, 1995.

[Kal03] E. Kaltofen. Polynomial factorization: a success story. In *ISSAC '03: Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 3–4. ACM Press, 2003.

[KS91] E. Kaltofen and B. D. Saunders. On Wiedemann's method of solving sparse linear systems. In H. F. Mattson, T. Mora, and T. R. N. Rao, editors, *Proceedings of AAECC-9*, volume 539 of *Lect. Notes Comput. Sci.*, pages 29–38. Springer-Verlag, 1991.

[Lan02] S. Lang. *Algebra*, volume 211 of *Graduate Texts in Mathematics*. Springer-Verlag, third edition, 2002.

[Lec06] G. Lecerf. Sharp precision in Hensel lifting for bivariate polynomial factorization. *Math. Comp.*, 75:921–933, 2006.

[Lec07a] G. Lecerf. Fast separable factorization and applications. Manuscript, Université de Versailles Saint-Quentin, France, 2007.

[Lec07b] G. Lecerf. Improved dense multivariate polynomial factorization algorithms. *J. Symbolic Comput.*, 42(4):477–494, 2007.

[Mag] The Magma computational algebra system for algebra, number theory and geometry. http://magma.maths.usyd.edu.au/magma/. Computational Algebra Group, School of Mathematics and Statistics, The University of Sydney, NSW 2006 Australia.

[Nie94] H. Niederreiter. Factoring polynomials over finite fields using differential equations and normal bases. *Math. Comp.*, 62(206):819–830, 1994.

[Ste05] A. Steel. Conquering inseparability: primary decomposition and multivariate factorization over algebraic function fields of positive characteristic. *J. Symbolic Comput.*, 40(3):1053–1075, 2005.

[Sto00] A. Storjohann. *Algorithms for matrix canonical forms*. PhD thesis, ETH, Zürich, Switzerland, 2000.

[Zip93] R. Zippel. *Effective Polynomial Computation.* Kluwer Academic Publishers, 1993.

Grégoire Lecerf, Laboratoire de Mathématiques (UMR 8100 CNRS), Université de Versailles Saint-Quentin, 45 avenue des États-Unis, 78035 Versailles, France

E-mail address: `Gregoire.Lecerf@math.uvsq.fr`