MATHEMATICS OF COMPUTATION
Volume 75, Number 254, Pages 921–933
S 0025-5718(06)01810-2
Article electronically published on January 3, 2006

SHARP PRECISION IN HENSEL LIFTING FOR BIVARIATE POLYNOMIAL FACTORIZATION

GRÉGOIRE LECERF

ABSTRACT. Popularized by Zassenhaus in the seventies, several algorithms for factoring polynomials use a so-called lifting and recombination scheme. Concerning bivariate polynomials, we present a new algorithm for the recombination stage that requires a lifting up to precision twice the total degree of the polynomial to be factored. Its cost is dominated by the computation of reduced echelon solution bases of linear systems. We show that our bound on precision is asymptotically optimal.

INTRODUCTION

Let F denote a polynomial in two variables x and y over a commutative field $\mathbb{K}$. All along this text F represents the polynomial we want to factor over $\mathbb{K}$. Many algorithms for factoring F proceed via so-called *lifting and recombination schemes*. Such schemes divide into three main stages. Informally speaking, in the first stage one of the two variables is specialized to a random value and the univariate polynomial obtained this way is factored. In the second stage, this factorization is *lifted* over a power series algebra and in the last stage the factorization over $\mathbb{K}$ is discovered from *recombinations* of the lifted factors. This article is devoted to the third stage only: we show that fast recombination is possible with lifting up to precision only twice the total degree d of F.

Recently, using the *logarithmic derivative method* introduced in [1], it has been proved that lifting up to a precision $d(d-1)+1$ is sufficient to efficiently recover the factors of F by means of linear algebra, whatever the characteristic of the base field is. This bound on precision is optimal for *small* positive characteristics [1, Remark 5.5]. If $\mathbb{K}$ has characteristic 0 or sufficiently *large*, a linear bound is sufficient, as first shown in [2].

On the other hand, in [12, 13], Ruppert introduced the idea of characterizing the absolute reducibility of F in terms of the existence of closed differential 1-forms $\omega = \frac{H}{F}dx + \frac{G}{F}dy$, where G and H are in $\mathbb{K}[x,y]$ and satisfy some constraints on their degrees. From this characterization, Gao derived an algorithm for computing both the absolute and rational factorizations of F [5]. The core of his algorithm relies on efficiently solving the linear system built from the condition of ω being

Received by the editor May 10, 2004 and, in revised form, February 28, 2005.
2000 *Mathematics Subject Classification.* Primary 12Y05, 68W30; Secondary 11Y16, 12D05, 13P05.
Key words and phrases. Polynomial factorization, Hensel lifting.

closed:

$$\frac{\partial}{\partial x}\left(\frac{G}{F}\right) = \frac{\partial}{\partial y}\left(\frac{H}{F}\right). \tag{1}$$

This equality can be rewritten in terms of the following crucial polynomial equation:

$$G\frac{\partial F}{\partial x} - F\frac{\partial G}{\partial x} = H\frac{\partial F}{\partial y} - F\frac{\partial H}{\partial y}. \tag{2}$$

In this article we combine both lifting and Gao's points of view. Our recombination algorithm is based on searching expressions of G and H in terms of the lifted factors. The only expression of G in terms of the lifted factors exactly corresponds to the logarithmic derivative method. The expression of H produces new linear equations that are the key to obtaining a sharp bound on precision via (2).

Notation. $\bar{\mathbb{K}}$ denotes the algebraic closure of $\mathbb{K}$. $\mathbb{K}[x,y]$ denotes the algebra of polynomials in two variables over $\mathbb{K}$, and $\mathbb{K}[x,y]_m$ the vector space of polynomials of degree at most m. The field of fractions of $\mathbb{K}[y]$ is denoted by $\mathbb{K}(y)$, the power series algebra over $\mathbb{K}$ is denoted by $\mathbb{K}[[x]]$, and its field of fractions by $\mathbb{K}((x))$. For any polynomial $G \in \mathbb{K}[x,y]$, the total degree of G is written $\deg(G)$, and its degree with respect to the variable x (resp. y) is written $\deg_x(G)$ (resp. $\deg_y(G)$). The resultant of F and G in $\mathbb{K}[x,y]$ with respect to y is denoted by $\mathrm{Res}_y(F,G)$. For compactness, the s-tuple $(\ell_1,\ldots,\ell_s)$ is represented by $\ell_{1:s}$, according to Gantmacher's notation. We also use the notation $\langle \mu_1,\ldots,\mu_r\rangle = \langle \mu_{1:r}\rangle$ to represent the $\mathbb{K}$-vector space generated by $\mu_{1:r}$.

Main results. We shall always work under the following assumptions:

$$\text{Hypothesis (H)} \quad \begin{cases} (i)\ \deg_y(F) = \deg(F) =: d; \\ (ii)\ \mathrm{Res}_y\left(F, \frac{\partial F}{\partial y}\right)(0) \neq 0. \end{cases}$$

Remark that (ii) implies $d \geq 1$. Hypothesis (H) is not really restrictive: if F is square-free, it can be ensured by means of a *generic* linear change of variables, but we will not discuss this question here. Hypothesis (i) corresponds to the fact that F is monic with respect to y. The monic (with respect to y) irreducible factors of F over $\mathbb{K}[x]$ (resp. over $\mathbb{K}[[x]]$) are then denoted by $F_{1:r}$ (resp. $\mathfrak{F}_{1:s}$). Of course we have $r \geq 1$, $s \geq 1$ and $s \geq r$. For convenience we will often use the *partial products*

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

To each $i \in \{1,\ldots,r\}$ we associate the vector $\mu_i \in \{0,1\}^s$, defined by $F_i = \prod_{j=1}^{s} \mathfrak{F}_j^{\mu_{i,j}}$. Since the μ_i have entries in $\{0,1\}$ and are pairwise orthogonal for the canonical dot product, up to a unique permutation, they form a reduced echelon basis. Throughout this text we assume that $\mu_{1:r}$ actually forms a reduced echelon basis.

The main idea of our new recombination algorithm relies in considering the following family of vector spaces parametrized by the precision $\sigma \geq 1$:

$$L_\sigma := \Big\{ (\ell_{1:s}, G, H) \in \mathbb{K}^s \times \mathbb{K}[x,y]_{d-1} \times \mathbb{K}[x,y]_{d-1} \mid$$
$$G - \sum_{i=1}^{s} \ell_i \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial y} \in (x,y)^\sigma,$$
$$H - \sum_{i=1}^{s} \ell_i \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial x} \in (x,y)^\sigma + (x^{\sigma-1}) \Big\}.$$

Here $(x,y)^\sigma$ represents the σth power of the ideal (x,y). It is generated by the monomials of total degree σ. The other ideal $(x,y)^\sigma + (x^{\sigma-1})$ is generated by the monomials of total degree σ plus the monomial $x^{\sigma-1}$. Let π denote the canonical projection from L_σ to $\mathbb{K}^s$. In Section 1 we prove the following theorem, which tells us that precision $\sigma = 2d$ is sufficient to recover the factorization of F by means of linear algebra:

Theorem 1. *Under Hypothesis (H), if $\sigma \geq 2d$ and if $\mathbb{K}$ has characteristic zero or at least $d(d-1)+1$, then*

$$\pi(L_\sigma) = \langle \mu_{1:r} \rangle. \tag{3}$$

In Section 2 we exhibit a family of examples, parametrized by the degree d, for which $\sigma \geq 2d-1$ is necessary in order to reach equality (3). This shows that the bound $2d$ given by the previous theorem is asymptotically optimal. Section 3 presents two algorithms for computing $\langle \mu_{1:r} \rangle$ from the $\mathfrak{F}_i$ given to precision σ. The first algorithm is deterministic and requires $\sigma \geq 2d$, whereas the second algorithm is probabilistic and faster but requires $\sigma \geq 2d+1$. Our new algorithms mainly reduce to computing reduced echelon solution bases of linear systems.

These new algorithms are faster than the ones given in [2] by a constant factor: respective precisions of the liftings are smaller and linear systems have fewer equations.

Related works. Concerning factorization of polynomials in general, we refer the reader to the classical books [21, 7], but also to the extensive bibliographies of [5, 8]. Here we only focus on lifting and recombination schemes and Gao's algorithm.

Factorization via Hensel lifting appeared first in the work of Zassenhaus at the end of the sixties for univariate polynomials with integer coefficients [20]. The idea of using lifting for $\mathbb{K}[x,y]$ is pioneered by [10, 19, 18]. For a long time the recombination step was performed by an exhaustive search, which means the computation of all the possible recombinations: true factors are recognized by means of Euclidean divisions. Of course, the cost of such a process is exponential in the number of lifted factors, hence in the total degree of the polynomial in the worst case.

Although polynomial time algorithms have been introduced in the eighties by Chistov, von zur Gathen, Grigoriev, Kaltofen and Lenstra (we refer the reader to the introduction of [5] for historical details and references), lifting and recombination schemes remain popular for they have been observed to be often faster in practice. Recently, Gao and Lauder proved that the average running time of such schemes is almost linear for bivariate polynomials over finite fields [6], which justifies the empirical observation.

In [16, 14, 15], T. Sasaki and his collaborators introduced the *zero-sum relations* method, also called the *trace recombination* method later: unfortunately no correct proof was given for ensuring polynomial time complexity in all cases (cf. the counterexample given in the introduction of [2]). Although the practical behavior of this method was very attractive, since it reduces the recombination stage to fast linear algebra computations, a valid polynomial bound on the required precision has remained unknown for a decade.

In a recent work [1], Belabas, van Hoeij, Klüners and Steel introduced the *logarithmic derivative method* for recombination. This method is mathematically equivalent to the recombination via traces [2, Section 2.3] since the logarithmic derivative corresponds to the generating series of the traces. The use of the logarithmic derivative allowed us to prove a quadratic bound on precision for Sasaki's algorithm, hence polynomial running time in *all cases*: for bivariate polynomials, precision $\sigma \geq d(d-1)+1$ is sufficient, whatever the characteristic of the base field. In addition, this bound is sharp for small positive characteristic.

This quadratic bound can be replaced by a linear one, namely $3d-2$ if the characteristic of the base field is zero or at least $d(d-1)+1$ as determined in [2]. In this article we modify the algorithm of [2]: we provide a new set of linear equations coming from the logarithmic derivatives with respect to both variables x and y. This way we reach the sharp precision $2d$. Based on these results, new probability estimates for multivariate polynomial factorization via Bertini's irreducibility theorem are presented in [9].

On the other hand, inspired by the work of Niederreiter [11] on factorization of univariate polynomials over finite fields and Ruppert's theorem on the characterization of absolute irreducible polynomials by means of differential 1-forms, Gao designed a factorization algorithm in [5] with the following feature: if the characteristic of the base field is zero or large enough, then the rational and absolute factorizations can be computed with $\mathcal{O}(d^5 \log(d)^{\mathcal{O}(1)})$ operations in $\mathbb{K}$, by means of a probabilistic algorithm. Under similar hypotheses, complexity $\mathcal{O}(d^\omega)$ is reached in [2] for rational factorization, also with a probabilistic method, where $\omega \leq 3$ denotes the *exponent of matrix multiplication complexity*. From the asymptotic point of view, we do not improve the complexity results of [2]: we show that our new algorithm belongs to the same complexity class but with better constants hidden behind the $\mathcal{O}$.

1. Proof of Theorem 1

This section is devoted to proving Theorem 1: the main idea consists in showing that conditions of Theorem 1 imply that G and H satisfy the closeness condition (2). We start with the easiest inclusion of (3):

Lemma 1. *Under Hypothesis (H), we have $\langle \mu_{1:r} \rangle \subseteq \pi(L_\sigma)$, for all $\sigma \geq 1$.*

Proof. Let $i \in \{1, \ldots, r\}$. Differentiating both sides of $F_i = \prod_{j=1}^s \mathfrak{F}_j^{\mu_{i,j}}$ with respect to y gives

$$\frac{\partial F_i}{\partial y} = \sum_{j=1}^s \mu_{i,j} \frac{\partial \mathfrak{F}_j}{\partial y} \prod_{k=1, k\neq j}^s \mathfrak{F}_k^{\mu_{i,k}}.$$

Multiplying both sides by $\hat{F}_i = \prod_{k=1}^{s} \mathfrak{F}_k^{1-\mu_{i,k}}$ yields

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

In a similar way, we obtain

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

Since both $\frac{\partial F_i}{\partial x}$ and $\frac{\partial F_i}{\partial y}$ have total degrees at most $\deg(F_i) - 1$, we deduce

$$\left(\mu_i, \hat{F}_i \frac{\partial F_i}{\partial y}, \hat{F}_i \frac{\partial F_i}{\partial x}\right) \in L_\sigma;$$

hence $\mu_i \in \pi(L_\sigma)$. □

The second lemma shows that the closeness condition (2) implies the nullity of the derivative of the residue of G/F at any root $\phi \in \bar{\mathbb{K}}[[x]]$ of $F(x, .)$. The proof follows that of [5, Lemma 2.4].

Lemma 2. *Under Hypothesis (H), let G and H be polynomials in $\bar{\mathbb{K}}[x, y]_{d-1}$ satisfying* (2)*, and let $\phi \in \bar{\mathbb{K}}[[x]]$ be such that $F(x, \phi) = 0$. We have:*

a. $\dfrac{d}{dx}\left(\dfrac{G(x,\phi)}{\frac{\partial F}{\partial y}(x,\phi)}\right) = 0$;

b. *If the characteristic of $\mathbb{K}$ is zero or at least $d(d-1)+1$, then*

$$\frac{G(x,\phi)}{\frac{\partial F}{\partial y}(x,\phi)} \in \bar{\mathbb{K}}.$$

Proof. According to Hypothesis (H), the polynomial F splits over $\bar{\mathbb{K}}[[x]]$. Let $\phi_1, \ldots, \phi_d$ denote the roots of F in $\bar{\mathbb{K}}[[x]]$ so that $F = \prod_{i=1}^{d}(y - \phi_i)$. For each $j \in \{1, \ldots, d\}$, we introduce

$$g_j := \frac{G(x,\phi_j)}{\frac{\partial F}{\partial y}(x,\phi_j)} \quad \text{and} \quad h_j := \frac{H(x,\phi_j)}{\frac{\partial F}{\partial y}(x,\phi_j)},$$

both belonging to $\bar{\mathbb{K}}[[x]]$. In $\bar{\mathbb{K}}((x))(y)$ the following identities hold, since G and H have degrees at most $d-1$ in y:

$$\frac{G}{F} = \sum_{j=1}^{d} \frac{g_j}{y - \phi_j}, \quad \frac{H}{F} = \sum_{j=1}^{d} \frac{h_j}{y - \phi_j}.$$

Differentiating the first equality with respect to x and the second one with respect to y, we obtain:

$$\frac{\partial}{\partial x}\left(\frac{G}{F}\right) = \sum_{j=1}^{d} \left(\frac{g_j}{(y-\phi_j)^2}\frac{d\phi_j}{dx} + \frac{1}{y-\phi_j}\frac{dg_j}{dx}\right),$$

$$\frac{\partial}{\partial y}\left(\frac{H}{F}\right) = -\sum_{j=1}^{d} \frac{h_j}{(y-\phi_j)^2}.$$

Using (1), we deduce

$$\frac{dg_j}{dx} = 0, \text{ for all } j \in \{1, \ldots, d\}. \tag{4}$$

This concludes part (a).

Let us now deal with part (b). Let p denote the characteristic of $\mathbb{K}$. If $p = 0$, then we deduce $g_j = g_j(0)$, from (4). Otherwise, if $p > 0$ we only deduce $g_j = g_j(0) + \mathcal{O}(x^p)$. Let $\bar{F}_j \in \bar{\mathbb{K}}[x, y]$ denote the irreducible factor of F that vanishes at ϕ_j. Consider the resultant $B(x) := \mathrm{Res}_y(\bar{F}_j, G - g_j(0)\frac{\partial F}{\partial y})$: according to basic properties on resultants, B has degree at most $(d-1)\deg(\bar{F}_j) \le d(d-1)$ and equals zero up to precision (x^p). According to the hypothesis on p, we deduce that $B = 0$. It follows that $\bar{F}_j$ divides $G - g_j(0)\frac{\partial F}{\partial y}$; hence $G(x, \phi_j) - g_j(0)\frac{\partial F}{\partial y}(x, \phi_j) = 0$, which yields $g_j = g_j(0)$ and concludes part (b) for positive characteristic. □

Proof of Theorem 1. According to Lemma 1, it remains to prove $\pi(L_\sigma) \subseteq \langle \mu_{1:r} \rangle$. Let $\ell_{1:s} \in \pi(L_\sigma)$. By construction, there exist G and H in $\mathbb{K}[x, y]_{d-1}$ such that

$$G - \sum_{i=1}^{s} \ell_i \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial y} \in (x, y)^\sigma, \quad H - \sum_{i=1}^{s} \ell_i \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial x} \in (x, y)^\sigma + (x^{\sigma-1}).$$

Differentiating the former equality with respect to x and the latter with respect to y yields

$$\frac{\partial G}{\partial x} - \sum_{i=1}^{s} \ell_i \left(\frac{\partial \hat{\mathfrak{F}}_i}{\partial x} \frac{\partial \mathfrak{F}_i}{\partial y} + \hat{\mathfrak{F}}_i \frac{\partial^2 \mathfrak{F}_i}{\partial xy} \right) \in (x, y)^{\sigma-1},$$
$$\frac{\partial H}{\partial y} - \sum_{i=1}^{s} \ell_i \left(\frac{\partial \hat{\mathfrak{F}}_i}{\partial y} \frac{\partial \mathfrak{F}_i}{\partial x} + \hat{\mathfrak{F}}_i \frac{\partial^2 \mathfrak{F}_i}{\partial xy} \right) \in (x, y)^{\sigma-1},$$

from which we deduce

$$G\frac{\partial F}{\partial x} - F\frac{\partial G}{\partial x} - \sum_{i=1}^{s} \ell_i \hat{\mathfrak{F}}_i \left(\frac{\partial \mathfrak{F}_i}{\partial y} \left(\frac{\partial F}{\partial x} - \mathfrak{F}_i \frac{\partial \hat{\mathfrak{F}}_i}{\partial x} \right) - F \frac{\partial^2 \mathfrak{F}_i}{\partial xy} \right) \in (x, y)^{\sigma-1},$$
$$H\frac{\partial F}{\partial y} - F\frac{\partial H}{\partial y} - \sum_{i=1}^{s} \ell_i \hat{\mathfrak{F}}_i \left(\frac{\partial \mathfrak{F}_i}{\partial x} \left(\frac{\partial F}{\partial y} - \mathfrak{F}_i \frac{\partial \hat{\mathfrak{F}}_i}{\partial y} \right) - F \frac{\partial^2 \mathfrak{F}_i}{\partial xy} \right) \in (x, y)^{\sigma-1}.$$

Then, using

$$\frac{\partial F}{\partial x} - \mathfrak{F}_i \frac{\partial \hat{\mathfrak{F}}_i}{\partial x} = \sum_{j=1}^{s} \hat{\mathfrak{F}}_j \frac{\partial \mathfrak{F}_j}{\partial x} - \sum_{j=1, j \ne i}^{s} \hat{\mathfrak{F}}_j \frac{\partial \mathfrak{F}_j}{\partial x} = \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial x},$$
$$\frac{\partial F}{\partial y} - \mathfrak{F}_i \frac{\partial \hat{\mathfrak{F}}_i}{\partial y} = \sum_{j=1}^{s} \hat{\mathfrak{F}}_j \frac{\partial \mathfrak{F}_j}{\partial y} - \sum_{j=1, j \ne i}^{s} \hat{\mathfrak{F}}_j \frac{\partial \mathfrak{F}_j}{\partial y} = \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial y},$$

we obtain

$$G\frac{\partial F}{\partial x} - F\frac{\partial G}{\partial x} - \left(H\frac{\partial F}{\partial y} - F\frac{\partial H}{\partial y} \right) \in (x, y)^{\sigma-1}.$$

From the assumption on σ and using the fact that $G\frac{\partial F}{\partial x} - F\frac{\partial G}{\partial x}$ and $H\frac{\partial F}{\partial y} - F\frac{\partial H}{\partial y}$ are polynomials of degrees at most $2d - 2$, the stronger equality (2) holds in $\mathbb{K}[[x]][y]$,

hence in $\mathbb{K}[x,y]$. Letting $i \in \{1,\dots,r\}$, for any j such that $\mu_{i,j} = 1$ we associate $\varphi_j \in \bar{\mathbb{K}}[[x]]$ such that $\mathfrak{F}_j(x,\varphi_j) = 0$. For such a j, Lemma 2 gives us $G(x,\varphi_j)/\frac{\partial F}{\partial y}(x,\varphi_j) \in \bar{\mathbb{K}}$. Letting $x = 0$, we have

$$G(0,y) - \sum_{i=1}^{s} \ell_i \hat{\mathfrak{F}}_i(0,y) \frac{\partial \mathfrak{F}_i}{\partial y}(0,y) \in (y)^\sigma,$$

and, using $\sigma \geq 2d \geq d$, it follows that

$$G(0,y) - \sum_{i=1}^{s} \ell_i \hat{\mathfrak{F}}_i(0,y) \frac{\partial \mathfrak{F}_i}{\partial y}(0,y) = 0.$$

Substituting $\varphi_j(0)$ for y in this equality yields

$$\frac{G(x,\varphi_j)}{\frac{\partial F}{\partial y}(x,\varphi_j)} = \frac{G(0,\varphi_j(0))}{\frac{\partial F}{\partial y}(0,\varphi_j(0))} = \ell_j.$$

It follows that F_i divides $G - \ell_j \frac{\partial F}{\partial y}$, and then, for any k such that $\mu_{i,k} = 1$, $G(x,\varphi_k) - \ell_j \frac{\partial F}{\partial y}(x,\varphi_k) = 0$. We deduce that $\ell_j = \ell_k$ and that $\ell_{1:s}$ belongs to $\langle \mu_{1:r} \rangle$, which concludes the proof. □

2. Lower bound on precision

In this section we show that $\sigma \geq 2d-1$ is necessary in order to ensure equality (3) of Theorem 1 in general. It follows that the precision $\sigma \geq 2d$ required by Theorem 1 is asymptotically sharp.

The lower bound $2d-1$ is realized by the following family of examples. We take $\mathbb{K} := \mathbb{C}$ (the field of complex numbers), $d \geq 2$, $F := y^d - y - x^{d-1}$. Let $\omega \in \mathbb{C}$ denote a $(d-1)$th primitive root of unity. Let $\ell_{1:s} := (1,\omega,\dots,\omega^{d-2},0)$, $G := (d-1)(y + x^{d-1})$, $H := (-(d-1)x^{d-2}y)$. Let $\phi_i \in \mathbb{K}[[x]]$, for $i \in \{1,\dots,d\}$, denote the roots of F, where

$$\phi_i = \omega^{i-1} + \frac{x^{d-1}}{d-1} + \mathcal{O}(x^{2d-2}), \quad \text{for } i \in \{1,\dots,d-1\}$$

and

$$\phi_d = -x^{d-1} + \mathcal{O}(x^{2d-2}).$$

According to our notation, we have $s = d$ and we let $\mathfrak{F}_i := y - \phi_i$, for $i \in \{1,\dots,s\}$. For $i \in \{1,\dots,d-1\}$, we compute

$$\begin{aligned}
\frac{G(x,\phi_i)}{\frac{\partial F}{\partial y}(x,\phi_i)} &= \frac{(d-1)\left(\omega^{i-1} + \frac{d}{d-1}x^{d-1}\right)}{d\left(\omega^{i-1} + \frac{x^{d-1}}{d-1}\right)^{d-1} - 1} + \mathcal{O}(x^{2d-2}) \\
&= \frac{(d-1)\left(\omega^{i-1} + \frac{d}{d-1}x^{d-1}\right)}{d\left((\omega^{i-1})^{d-1} + (\omega^{i-1})^{d-2}\, x^{d-1}\right) - 1} + \mathcal{O}(x^{2d-2}) \\
&= \frac{(d-1)\omega^{i-1} + dx^{d-1}}{d-1+d\omega^{-(i-1)}x^{d-1}} + \mathcal{O}(x^{2d-2}) \\
&= \omega^{i-1} + \mathcal{O}(x^{2d-2}).
\end{aligned}$$

We deduce

$$G - \sum_{i=1}^{s} \ell_i \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial y} \in (x^{2d-2}). \tag{5}$$

Multiplying both sides of this equality by $-x^{d-2} = \frac{\partial \mathfrak{F}_i}{\partial x} + \mathcal{O}(x^{2d-3})$ and using $\frac{\partial \mathfrak{F}_i}{\partial y} = 1$, for $i \in \{1, \ldots, d-1\}$, we deduce

$$-x^{d-2}G - \sum_{i=1}^{s} \ell_i \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial x} \in (x^{2d-3}),$$

and, using $x^{d-2}G + H \in (x^{2d-3})$, we finally get

$$H - \sum_{i=1}^{s} \ell_i \hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial x} \in (x^{2d-3}). \tag{6}$$

By combining (5) and (6), we deduce that $(\ell_{1:s}, G, H) \in L_{2d-2}$; hence $\ell_{1:s} \in \pi(L_{2d-2})$.

Consider i such that $\mu_{i,d} = 1$. If we had $\pi(L_{2d-2}) = \langle \mu_{1:r} \rangle$ this would imply $\mu_{i,j} = 0$ for any $j \neq d$; hence $\mathfrak{F}_d$ would belong to $\mathbb{K}[x, y]$, which is not possible. It follows that $\sigma \geq 2d - 1$ is necessary in order to ensure (3). It is worth mentioning that F is irreducible (over $\mathbb{C}$): for instance, this comes from the Eisenstein-Dumas criterion (see [4] for recent advances in this topic).

3. Recombination algorithms

In this section we treat the problem of computing $\mu_{1:r}$ from $\mathfrak{F}_{1:s}$ known up to a certain precision. According to our notation and assumptions, if $\pi(L_\sigma) = \langle \mu_{1:r} \rangle$ holds, then $\mu_{1:r}$ equals the reduced echelon basis of $\pi(L_\sigma)$. *From now on, we assume that* $\mathbb{K}$ *has either characteristic* 0 *or at least* $d(d-1)+1$.

We are going to describe two algorithms for computing the μ_i that are adapted from [2]. The first one is deterministic and directly exploits Theorem 1. The second one is probabilistic and mainly gains a factor of d in complexities. We start this section with some preliminaries about the complexity model we use. Concerning the complexities of the lifting stage and the computation of the F_i from the μ_i, we refer the reader to [2].

Complexity model. For our complexity analysis, we use the *computation tree* model [3, Chapter 4] from the *total complexity* point of view. This means that complexity estimates charge a constant cost for each arithmetic operation ($+$, $-$, $\times$, $\div$) and the equality test. All the constants in the base fields (or rings) of the trees are thought to be freely at our disposal.

Polynomials and series are represented by dense vectors of their coefficients in the canonical monomial basis. 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 [7, Chapter 8.3], for any positive integers d_1 and d_2, we assume that M satisfies: $M(d_1 d_2) \leq d_1^2 \mathsf{M}(d_2)$, and $\mathsf{M}(d_2)/d_2 \geq \mathsf{M}(d_1)/d_1$ if $d_2 \geq d_1$. In particular, this implies the *super-additivity* of M, that is: $\mathsf{M}(d_1 + d_2) \geq \mathsf{M}(d_1) + \mathsf{M}(d_2)$ for any positive integers d_1 and d_2.

The constant ω denotes a feasible matrix multiplication exponent as in [7, Chapter 12], so that two $n \times n$ matrices over $\mathbb{K}$ can be multiplied with $\mathcal{O}(n^\omega)$ field operations. As in [17], we assume that $2 < \omega \leq 3$. We recall the following complexity for linear system solving, which is a corollary of [17, Theorem 2.10]:

Lemma 3. *The computation of the reduced echelon solution basis of a linear system over $\mathbb{K}$ with s unknowns and $m \geq s$ equations requires $\mathcal{O}(ms^{\omega-1})$ operations in $\mathbb{K}$.*

In this section we shall use the notation $\operatorname{coeff}\left(G, x^j y^k\right)$, that denotes the coefficient of $x^j y^k$ in $G \in \mathbb{K}[[x]][[y]]$.

Deterministic recombination algorithm. Assume we are given $\mathfrak{F}_1, \ldots, \mathfrak{F}_s$ to precision (x^σ). Our aim is to compute the reduced echelon basis of $\pi(L_\sigma)$. For this purpose, we use the following linear system D_σ, with s unknowns $\ell_{1:s}$:

$$D_\sigma \begin{cases} \sum_{i=1}^s \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial y}, x^j y^k\right) = 0, \ k \leq d-1, \ d \leq j+k \leq \sigma - 1, \\ \sum_{i=1}^s \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial x}, x^j y^k\right) = 0, \ k \leq d-1, \ j \leq \sigma - 2, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad d \leq j+k \leq \sigma - 1. \end{cases}$$

D_σ is related to L_σ as follows:

Lemma 4. *Under Hypothesis (H), for all $\sigma \geq d$, we have*

$$\pi(L_\sigma) = \{\ell_{1:s} \in \mathbb{K}^s \mid D_\sigma\}.$$

Proof. The linear system D_σ is directly built from $\pi(L_\sigma)$: by construction, $\mathfrak{F}_i$ is monic with respect to y, which implies that $\frac{\partial \mathfrak{F}_i}{\partial x}$ has degree in y at most $\deg_y(\mathfrak{F}_i) - 1$. It follows that both $\hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial y}$ and $\hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial x}$ have degrees at most $d-1$ in y, which justifies the restriction $k \leq d-1$ in the construction of D_σ. The first set of equations of D_σ runs over the monomials $x^j y^k$ that belong neither to $(x,y)^\sigma$ nor to $\mathbb{K}[x,y]_{d-1}$. The second set of equations runs over the monomials $x^j y^k$ that belong neither to $(x,y)^\sigma + (x^{\sigma-1})$ nor to $\mathbb{K}[x,y]_{d-1}$. □

Here follows the first algorithm together with its complexity analysis.

Algorithm Recombination
- Input: $\mathfrak{F}_{1:s}$ to precision (x^σ).
- Output: $\mu_{1:r}$.
  - a. For each $i \in \{1, \ldots, s\}$, compute $\hat{\mathfrak{F}}_i$ as the quotient of F by $\mathfrak{F}_i$, using Euclidean divisions with respect to y to precision (x^σ).
  - b. Compute $(\hat{\mathfrak{F}}_1 \frac{\partial \mathfrak{F}_1}{\partial y}, \ldots, \hat{\mathfrak{F}}_s \frac{\partial \mathfrak{F}_s}{\partial y})$ to precision (x^σ).
  - c. Compute $(\hat{\mathfrak{F}}_1 \frac{\partial \mathfrak{F}_1}{\partial x}, \ldots, \hat{\mathfrak{F}}_s \frac{\partial \mathfrak{F}_s}{\partial x})$ to precision $(x^{\sigma-1})$.
  - d. Compute the reduced echelon solution basis of D_σ.

Proposition 1. *Under Hypothesis (H), for $\sigma = 2d$, Algorithm* Recombination *is correct and requires $\mathcal{O}(\mathsf{M}(\sigma)\mathsf{M}(d)s + \sigma d s^{\omega-1}) \subseteq \mathcal{O}(d^{\omega+1} + d\mathsf{M}(d)^2)$ operations in $\mathbb{K}$.*

Proof. The correctness directly follows from the combination of the previous lemma and Theorem 1. The Euclidean divisions are well defined since the $\mathfrak{F}_i$ are monic with respect to y. Step a costs $\mathcal{O}(\mathsf{M}(\sigma)\mathsf{M}(d)s)$. The costs of steps b and c belong to $\mathcal{O}(\mathsf{M}(\sigma)\mathsf{M}(d)s)$. The linear system D_σ has s unknowns and $\mathcal{O}(\sigma d)$ equations. Thus the last step costs $\mathcal{O}(\sigma d s^{\omega-1})$ operations, by Lemma 3. Lastly, the right-hand side of the inclusion follows from $s \leq d$. □

The asymptotic cost of this algorithm is roughly the same as the one of [2, Section 2.2] but it gains two constant factors. The first one comes from using precision $2d$ instead of $3d-2$. The second one concerns the size of the linear system: both have the same number s of unknowns but ours has $2d^2-1$ equations compared to $\frac{5}{2}d(d-1)$.

Probabilistic recombination algorithm. We now detail a faster probabilistic algorithm, by showing that y may be replaced by ux, for two random values $u \in \mathbb{K}$, with a high probability of success. This leads to a linear system with fewer equations than D_σ, reducing the cost of the linear algebra stage mainly by a factor of d. The same factor also concerns other steps since the instantiations of y to ux can be done at the beginning of the process. A crucial advantage of this technique is that it avoids constructing D_σ at all. The only slight drawback versus the previous algorithm is that the required precision is $2d+1$ instead of $2d$. In order to avoid confusion we use τ to denote this precision instead of σ.

For any $u \in \mathbb{K}$, we introduce the following linear system P_τ^u:

$$P_\tau^u \begin{cases} \sum_{i=1}^s \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}_i(x,ux)\frac{\partial \mathfrak{F}_i}{\partial x}(x,ux), x^j\right) = 0, \ d \le j \le \tau-2, \\ \sum_{i=1}^s \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}_i(x,ux)\frac{\partial \mathfrak{F}_i}{\partial y}(x,ux), x^j\right) = 0, \ d \le j \le \tau-2. \end{cases}$$

The probabilistic algorithm for recombination proceeds as follows:

Algorithm ProbabilisticRecombination

Input: $\mathfrak{F}_{1:s}$ to precision (x^τ), a and b in $\mathbb{K}$.
Output: $\mu_{1:r}$.

a. For each $i \in \{1,\dots,s\}$ and each $u \in \{a,b\}$, compute $f_i^u := \mathfrak{F}_i(x,ux)$, $g_i^u := \frac{\partial \mathfrak{F}_i}{\partial y}(x,ux)$ to precision (x^τ) and $h_i^u := \frac{\partial \mathfrak{F}_i}{\partial x}(x,ux)$ to precision $(x^{\tau-1})$.
b. For each $u \in \{a,b\}$, let $A_1^u := 1$, $B_s^u := 1$.
For each i from 2 to s and each $u \in \{a,b\}$, compute $A_i^u := A_{i-1}^u f_{i-1}^u$, $B_{s-i+1}^u := B_{s-i+2}^u f_{s-i+2}^u$ to precision (x^τ).
c. For each $i \in \{1,\dots,s\}$ and each $u \in \{a,b\}$, compute $\hat{\mathfrak{F}}_i(x,ux)\frac{\partial \mathfrak{F}_i}{\partial y}(x,ux)$ as $g_i^u A_i^u B_i^u$ to precision (x^τ) and $\hat{\mathfrak{F}}_i(x,ux)\frac{\partial \mathfrak{F}_i}{\partial x}(x,ux)$ as $h_i^u A_i^u B_i^u$ to precision $(x^{\tau-1})$ (since $A_i^u B_i^u = \prod_{j=1, j\neq i}^s f_j^u$).
d. Return the reduced echelon solution basis of the union of P_τ^a and P_τ^b.

We start with the complexity analysis.

Proposition 2. *Under Hypothesis (H), for $\tau = 2d+1$, Algorithm* ProbabilisticRecombination *requires $\mathcal{O}(d\tau + \mathsf{M}(\tau)s + \tau s^{\omega-1}) \subseteq \mathcal{O}(\mathsf{M}(d)d + d^\omega)$ operations in $\mathbb{K}$.*

Proof. Step a performs $\mathcal{O}(d\tau)$ operations. The total cost of step b belongs to $\mathcal{O}(\mathsf{M}(\tau)s)$. Step c costs $\mathcal{O}(\mathsf{M}(\tau)s)$. The final join system of P_τ^a and P_τ^b has s unknowns and $\mathcal{O}(\tau)$ equations. Thus the cost of step d comes from Lemma 3. Lastly, the right-hand side of the inclusion follows from $s \le d$. □

In order to prove the correctness and study the probability of success of this algorithm, we introduce the following vector space:

$$\Lambda_\tau := \Bigg\{ \ell_{1:s} \in \mathbb{K}^s \mid \sum_{i=1}^{s} \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial x}, x^j y^k\right) = 0, \quad k \le d-1,\ d \le j+k \le \tau-2, \quad \sum_{i=1}^{s} \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}_i \frac{\partial \mathfrak{F}_i}{\partial y}, x^j y^k\right) = 0, \quad k \le d-1,\ d \le j+k \le \tau-2 \Bigg\}.$$

From Lemma 4 it is easy to see that

$$\Lambda_\tau \subseteq \pi(L_{\tau-1}). \tag{7}$$

Let z denote a new variable. Substituting xz for y, we obtain:

$$\Lambda_\tau = \Bigg\{ \ell_{1:s} \in \mathbb{K}^s \mid \sum_{i=1}^{s} \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}_i(x,xz) \frac{\partial \mathfrak{F}_i}{\partial x}(x,xz), x^j z^k\right) = 0, \quad k \le d-1,\ d \le j \le \tau-2, \quad \sum_{i=1}^{s} \ell_i \operatorname{coeff}\left(\hat{\mathfrak{F}}_i(x,xz) \frac{\partial \mathfrak{F}_i}{\partial y}(x,xz), x^j z^k\right) = 0, \quad k \le d-1,\ d \le j \le \tau-2 \Bigg\}.$$

For any $u \in \mathbb{K}$, we shall use $\Lambda_\tau^u := \{\ell_{1:s} \in \mathbb{K}^s \mid P_\tau^u\}$. The following technical lemma tells us that specializing u to two different values a and b in $\mathbb{K}$ allows us to recover Λ_τ by means of solving the join system $P_\tau^a \cup P_\tau^b$, except if (a,b) belongs to a certain proper Zariski closed subset of $\mathbb{K}^2$.

Lemma 5. *For any $b \in \mathbb{K}$, there exists a nonzero polynomial $\mathcal{P}_b \in \mathbb{K}[z]$ of degree at most $(\dim(\Lambda_\tau^b) - \dim(\Lambda_\tau))(d-1)$ such that $\mathcal{P}_b(a) \neq 0$ implies $\Lambda_\tau = \Lambda_\tau^a \cap \Lambda_\tau^b$.*

Proof. Obviously, we have $\Lambda_\tau \subseteq \Lambda_\tau^a \cap \Lambda_\tau^b$, for any a and b. Let $\lambda_{1:\dim(\Lambda_\tau^b)}$ be a basis of Λ_τ^b such that the $\dim(\Lambda_\tau)$ first vectors form a basis of Λ_τ. For any $\lambda_k \notin \Lambda_\tau$ there exists $j \in \{d, \ldots, \tau-2\}$ such that one polynomial among

$$\sum_{i=1}^{s} \lambda_{k,i} \operatorname{coeff}\left(\hat{\mathfrak{F}}_i(x,zx) \frac{\partial \mathfrak{F}_i}{\partial y}(x,zx), x^j\right)$$

and

$$\sum_{i=1}^{s} \lambda_{k,i} \operatorname{coeff}\left(\hat{\mathfrak{F}}_i(x,zx) \frac{\partial \mathfrak{F}_i}{\partial x}(x,zx), x^j\right)$$

is not zero: let $p_k(z)$ be one of them that is not zero. We take $\mathcal{P}_b$ as the product of all such p_k. By construction, each p_k has degree at most $d-1$. Finally for any $a \in \mathbb{K}$ such that $\mathcal{P}_b(a) \neq 0$, we have $\lambda_k \notin \Lambda_\tau^a$. □

We are now ready to show the correctness of the probabilistic recombination algorithm:

Proposition 3. *Under Hypothesis (H), if $\tau = 2d+1$, then for any a and b in $\mathbb{K}$ such that $\mathcal{P}_b(a) \neq 0$ (where $\mathcal{P}_b$ is the polynomial defined in the previous lemma) Algorithm* ProbabilisticRecombination *is correct.*

Proof. According to the previous lemma, we have $\Lambda_\tau = \Lambda_\tau^a \cap \Lambda_\tau^b$. Thus, from (7) and Theorem 1, the equality $\Lambda_\tau^a \cap \Lambda_\tau^b = \langle \mu_{1:r} \rangle$ follows. □

Bad choices of a and b will result in wrong factors F_i. Such situations can be easily detected by computing the F_i candidates and performing Euclidean division of F. Indeed, using [2, Proposition 4] we know that the μ_i returned by ProbabilisticRecombination are correct if and only if they all have entries in $\{0,1\}$ (this requires the characteristic to be either zero or at least d). In addition, all the computed μ_i that have entries in $\{0,1\}$ correspond to true factors but not necessarily irreducible. This allows one to split the original factorization problem into smaller problems. Furthermore, according to our assumption on the characteristic of $\mathbb{K}$ and $\deg(\mathcal{P}_b) \leq d(d-1)$, for any b it is always possible to find $a \in \{0, \ldots, d(d-1)\}$ such that $\mathcal{P}_b(a) \neq 0$. This way, the probabilistic strategy can be used in order to always return correct results.

Lastly, as in the deterministic algorithm, we gain constant factors in the precision of the series and the size of the linear system compared to [2, Section 2]: our precision $2d+1$ is to be compared to $4d-3$, and our linear system $P_\tau^a \cup P_\tau^b$ contains $4d$ equations compared to $6(d-1)$.

Acknowledgments

I thank Arne Storjohann for pointing out that the complexity of the reduced echelon computation used in [2, Section 2.2] could be improved thanks to [17, Theorem 2.10]. I am also grateful to an anonymous referee for his useful comments.

References

[1] K. Belabas, M. van Hoeij, J. Klüners, and A. Steel, *Factoring polynomials over global fields*, Manuscript, October 2004.

[2] A. Bostan, G. Lecerf, B. Salvy, É. Schost, and B. Wiebelt, *Complexity issues in bivariate polynomial factorization*, Proceedings of ISSAC 2004, ACM Press, 2004, pp. 42–49. MR2126923 MR1440179 (99c:68002)

[3] P. Bürgisser, M. Clausen, and M. A. Shokrollahi, *Algebraic complexity theory*, Springer-Verlag, 1997. MR1816701 (2002f:52013)

[4] S. Gao, *Absolute irreducibility of polynomials via Newton polytopes*, J. Algebra **237** (2001), no. 2, 501–520.

[5] ———, *Factoring multivariate polynomials via partial differential equations*, Math. Comp. **72** (2003), 801–822. MR1954969 (2003m:12014)

[6] S. Gao and A. G. B. Lauder, *Hensel lifting and bivariate polynomial factorisation over finite fields*, Math. Comp. **71** (2002), no. 240, 1663–1676. MR1933049 (2003j:11149)

[7] J. von zur Gathen and J. Gerhard, *Modern computer algebra*, second ed., Cambridge University Press, 2003. MR2001757 (2004g:68202)

[8] E. Kaltofen, *Polynomial factorization: A success story*, Proceedings of ISSAC 2003, ACM Press, 2003, pp. 3–4.

[9] G. Lecerf, *Improved dense multivariate polynomial factorization algorithms*, Manuscript, January 2005.

[10] D. R. Musser, *Multivariate polynomial factorization*, J. Assoc. Comput. Mach. **22** (1975), 291–308. MR0396470 (53:335a)

[11] H. Niederreiter, *A new efficient factorization algorithm for polynomials over small finite fields*, Appl. Algebra Engrg. Comm. Comput. **4** (1993), no. 2, 81–87. MR1223850 (94h:11112)

[12] W. M. Ruppert, *Reduzibilität ebener Kurven*, J. Reine Angew. Math. **369** (1986), 167–191. MR0850633 (88j:14010)

[13] ———, *Reducibility of polynomials $f(x, y)$ modulo p*, J. Number Theory **77** (1999), no. 1, 62–70. MR1695700 (2000d:11128)

[14] T. Sasaki, T. Saito, and T. Hilano, *Analysis of approximate factorization algorithm. I*, Japan J. Indust. Appl. Math. **9** (1992), no. 3, 351–368. MR1189944 (94a:12002)

[15] T. Sasaki and M. Sasaki, *A unified method for multivariate polynomial factorizations*, Japan J. Indust. Appl. Math. **10** (1993), no. 1, 21–39. MR1208180 (94a:13029)

[16] T. Sasaki, M. Suzuki, M. Kolář, and M. Sasaki, *Approximate factorization of multivariate polynomials and absolute irreducibility testing*, Japan J. Indust. Appl. Math. **8** (1991), no. 3, 357–375. MR1137647 (92j:12002)

[17] A. Storjohann, *Algorithms for matrix canonical forms*, Ph.D. thesis, ETH, Zürich, 2000, http://www.scg.uwaterloo.ca/~astorjoh.

[18] P. S. Wang, *An improved multivariate polynomial factoring algorithm*, Math. Comp. **32** (1978), no. 144, 1215–1231. MR0568284 (58:27887b)

[19] P. S. Wang and L. P. Rothschild, *Factoring multivariate polynomials over the integers*, Math. Comp. **29** (1975), 935–950. MR0396471 (53:335b)

[20] H. Zassenhaus, *On Hensel factorization I*, J. Number Theory **1** (1969), no. 1, 291–311. MR0242793 (39:4120)

[21] R. Zippel, *Effective polynomial computation*, Kluwer Academic Publishers, 1993.

Laboratoire de mathématiques (UMR 8100 CNRS), université de Versailles Saint-Quentin-en-Yvelines, 45 avenue des États-Unis, 78035 Versailles, France

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