Computing the multilinear factors of lacunary polynomials without heights

Arkadev Chattopadhyay* Bruno Grenet† Pascal Koiran‡
Natacha Portier‡ Yann Strozecki§

Abstract

We present a deterministic algorithm which computes the multilinear factors of multivariate lacunary polynomials over number fields. Its complexity is polynomial in ℓ^n where ℓ is the lacunary size of the input polynomial and n its number of variables, that is in particular polynomial in the logarithm of its degree. We also provide a randomized algorithm for the same problem of complexity polynomial in ℓ and n.

Over other fields of characteristic zero and finite fields of large characteristic, our algorithms compute the multilinear factors having at least three monomials of multivariate polynomials. Lower bounds are provided to explain the limitations of our algorithm. As a by-product, we also design polynomial-time deterministic polynomial identity tests for families of polynomials which were not known to admit any.

Our results are based on so-called Gap Theorem which reduce high-degree factorization to repeated low-degree factorizations. While previous algorithms used Gap Theorems expressed in terms of the heights of the coefficients, our Gap Theorems only depend on the exponents of the polynomials. This makes our algorithms more elementary and general, and faster in most cases.

*School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai, India; arkadev.c@tifr.res.in.

†LIRMM, Univ. Montpellier, CNRS, Montpellier, France; bruno.grenet@lirmm.fr.

‡Univ Lyon, EnsL, UCBL, CNRS, LIP, F-69342, LYON Cedex 07, France; [pascal.koiran,natacha.portier]@ens-lyon.fr.

§DAVID laboratory, Université de Versailles Saint-Quentin, France; yann.strozecki@uvsq.fr.

1 Introduction

The *lacunary representation* of a polynomial

$$P(X_1,\dots,X_n)=\sum_{j=1}^{k} a_j X_1^{\alpha_{1,j}}\cdots X_n^{\alpha_{n,j}}$$

is the list of the tuples $(a_j,\alpha_{1,j},\dots,\alpha_{n,j})$ for $1\le j\le k$. This representation, sometimes called *sparse* or *supersparse representation* in the literature, allows very high degree polynomials to be represented in a concise manner. The factorization of lacunary polynomials has been investigated in a series of papers. Cucker et al. [11] first proved that integer roots of univariate integer lacunary polynomials can be found in polynomial time. This result was generalized by Lenstra [35] who proved that low-degree factors of univariate lacunary polynomials over algebraic number fields can also be found in polynomial time. More recently, Kaltofen and Koiran [24, 25] generalized Lenstra's results to bivariate and then multivariate lacunary polynomials. A common point to these algorithms is that they all rely on a so-called *Gap Theorem*: If F is a factor of $P(X)=\sum_{j=1}^{k}a_jX^{\alpha_j}$, then there exists ℓ such that F is a factor of both $\sum_{j=1}^{\ell}a_jX^{\alpha_j}$ and $\sum_{j=\ell+1}^{k}a_jX^{\alpha_j}$. (Here, X is a vector of variables of length at least 1, and the α_j's are vectors of exponents.) Moreover, the different Gap Theorems in these papers are all based on the notion of height of an algebraic number, and some of them use quite sophisticated results of number theory.

In this paper, we are interested in more elementary proofs for these results. We first focus on the case of linear factors of bivariate lacunary polynomials as Kaltofen and Koiran [24]. Yet unlike their result our algorithm works over number fields, and is extended to multivariate polynomials, and to the computation of multilinear factors, with multiplicities. This was investigated by Kaltofen and Koiran [25], who also dealt with low-degree factors. Grenet [16, 18] has generalized our approach to low-degree factors, subsuming all known previous results.

We prove a new Gap Theorem that does not depend on the height of an algebraic number but only on the exponents. In particular, our Gap Theorem is valid for any field of characteristic zero and we also extend it to the case of multilinear factors of multivariate polynomials. As a result, we get a new, more elementary algorithm for finding multilinear factors of multivariate lacunary polynomials over an algebraic number field. In particular, this new algorithm is easier to implement (see [17] for a recent implementation) since there is no need to explicitly compute some constants from number theory, and the use of the Gap Theorem does not require the evaluation of the heights of the coefficients of the polynomial. We also compute the multiplicities of the factors. With our method this comes for free, which makes our algorithm faster than the previous ones in most cases (see Section 4.5 for details).

Our algorithm can also be used for absolute factorization, that is the factorization of a polynomial in an algebraic closure of the field generated by its coefficients. More precisely, it can be used to compute in polynomial time the multilinear factors with at least three monomials of a lacunary multivariate polynomial. Note that univariate factorization reduces to the computation of binomial factors. And since the absolute factorization of a univariate polynomial of degree d is a product of d linear factors, these factors cannot even be listed in polynomial time. We shall also discuss the application of our algorithms to other fields of characteristic zero.

We use the same methods to prove a Gap Theorem for polynomials over some fields of positive characteristic, yielding an algorithm to find multilinear factors of multivariate lacunary polynomials with at least three monomials. We show that detecting the existence of binomial factors, that is factors with exactly two monomials, is NP-hard. This follows from the fact that finding univariate linear factors over finite fields is NP-hard [31, 2, 26]. In algebraic number fields we can find *all* multilinear factors in polynomial time, even the binomial ones. For this we rely, as Kaltofen and Koiran did, on Lenstra's univariate algorithm [35].

Our Gap Theorems are based on the *valuation* of a univariate polynomial, that is the maximum integer v such that X^v divides the polynomial. We give an upper bound on the valuation of a nonzero polynomial

$$P(X) = \sum_{j=1}^{k} a_j X^{\alpha_j} (vX + t)^{\beta_j} (uX + w)^{\gamma_j}.$$

This bound can be viewed as an extension of a result due to Hajós [21, see also 37]. We also note that Kayal and Saha [30] recently used the valuation of square roots of polynomials to make some progress on the "Sum of Square Roots" problem.

Lacunary polynomials have been studied with respect to other computational tasks. For instance, Plaisted [39] showed the NP-hardness of computing the greatest common divisor (gcd) of two univariate integer lacunary polynomials. His results were extended to prove that testing the irreducibility of a lacunary polynomial is NP-hard for polynomials with coefficients in $\mathbb{Z}$ or in a finite field [43, 29, 24]. On the other hand, some efficient algorithms for lacunary polynomials have also been given, for instance for testing if a polynomial vanishes on roots of unity [8], if a polynomial has a real root [3], or for the detection of perfect powers [14, 15] or interpolation [27].

Another approach for computing with lacunary polynomials is to give algorithms with a polynomial complexity in the logarithm of the degree (that is in the size of the exponents) but not in the number of terms or the size of the coefficients. This approach has been used to circumvent Plaisted's NP-hardness result on the gcd [12, 1].

Note that for all the problems we address, there exist algorithms with a polynomial complexity in the degree of the polynomials. They are used as subroutines of our algorithms. We refer the reader to [42] for details and references on these algorithms.

A preliminary version of this paper was published in the conference ISSAC 2013 [7] that contains the bivariate case of our results. The present paper gives more details on the algorithms especially for the computation of the multiplicities of the factors, and the generalization to multivariate polynomials is new. In positive characteristic, it includes a more general NP-hardness result. We also give a new Polynomial Identity Testing algorithm for sums of products of dense polynomials.

Organization of the paper

Section 2 is devoted to our main technical results. In Section 2.1, we give a bound on the valuation of a nonzero polynomial $P = \sum_j a_j X^{\alpha_j}(uX+v)^{\beta_j}$ over a field of characteristic 0. This result is extended in Section 2.2 to polynomials of the form $\sum_j a_j \prod_i f_i^{\alpha_{ij}}$ where the f_i's are low-degree polynomials. In Section 2.3, we discuss the tightness of these results.

In Section 3, we use these valuation bounds to get new Gap Theorems, respectively adapted to linear and multilinear factors. We also give in Section 3.2 a first application of these Gap Theorems: We give polynomial identity testing algorithms for the above mentioned families of polynomials.

Section 4 presents our main application: the factorization of lacunary polynomials. We begin with the computation of linear factors of bivariate polynomials over a number field in Section 4.1, and then extend it to multilinear factors in Section 4.2. Then, we generalize these algorithms to multivariate polynomials in Section 4.3. We briefly discuss absolute factorization and factorization in other fields of characteristic zero in Section 4.4. Section 4.5 presents a comparison between our techniques and existing ones.

Finally, the case of positive characteristic is investigated in Section 5. We show how to partially extend the results of Section 2 to positive characteristic, and we give similar algorithms as in Section 4. We note that these algorithms are less general, but we also give NP-hardness results explaining this lack of generality.

To understand the basics of our method, one can focus on the computation of linear factors of bivariate polynomials over a number field. To this end, one only has to read Section 2.1, Theorem 3.1 and its proof in Section 3, and Section 4.1.

Acknowledgments

We wish to thank Sébastien Tavenas for his help on Proposition 2.8, and Erich L. Kaltofen for pointing us out a mistake in a previous version of Theorem 5.4. We are grateful to the two anonymous reviewers for very interesting remarks and suggestions that improved the presentation of this paper.

2 Wronskian and valuation

In this section, we consider a field $\mathbb{K}$ of characteristic zero and polynomials over $\mathbb{K}$. We denote by $\operatorname{val}(P)$ the valuation of a univariate polynomial P over $\mathbb{K}$, that is the largest v such that X^v divides P. More generally, we denote by $\mu_F(P)$ the *multiplicity of F as a factor of P*, that is the largest m such that F^m divides P. In particular, $\operatorname{val}(P) = \mu_X(P)$ for $P \in \mathbb{K}[X]$.

2.1 Valuation upper bound

Theorem 2.1. *Let $P = \sum_{j=1}^{\ell} a_j X^{\alpha_j}(uX+v)^{\beta_j}$ with $\alpha_1 \le \cdots \le \alpha_\ell$ and $uv \neq 0$. If P is nonzero, its valuation is at most $\max_j(\alpha_j + \binom{\ell+1-j}{2})$.*

Our proof of Theorem 2.1 is based on the so-called *Wronskian* of a family of polynomials. This is a classical tool for the study of differential equations but it has also been used in the field of algebraic complexity [20, 33, 30, 32, 13].

Definition 2.2. Let $f_1, \ldots, f_\ell \in \mathbb{K}[X]$. Their *Wronskian* is the determinant of the *Wronskian matrix*

$$\operatorname{wr}(f_1,\ldots,f_\ell) = \det \begin{bmatrix} f_1 & f_2 & \cdots & f_\ell \\ f_1' & f_2' & \cdots & f_\ell' \\ \vdots & \vdots & & \vdots \\ f_1^{(\ell-1)} & f_2^{(\ell-1)} & \cdots & f_\ell^{(\ell-1)} \end{bmatrix}.$$

The main property of the Wronskian is its relation to linear independence. The following result is classical (see [5] for a simple proof of this fact).

Proposition 2.3. *The Wronskian of any polynomials $f_1, \ldots, f_\ell \in \mathbb{K}[X]$ is nonzero if and only if the f_j's are linearly independent over $\mathbb{K}$.*

To prove Theorem 2.1, an easy lemma on the valuation of the Wronskian is needed.

Lemma 2.4. *Let $f_1, \ldots, f_\ell \in \mathbb{K}[X]$. Then*

$$\operatorname{val}(\operatorname{wr}(f_1,\ldots,f_\ell)) \ge \sum_{j=1}^{\ell} \operatorname{val}(f_j) - \binom{\ell}{2}.$$

Proof. Each term of the determinant is a product of ℓ terms, one from each column and one from each row. The valuation of such a term is at least $\sum_j \mathrm{val}(f_j) - \sum_{i=1}^{\ell-1} i$ since for all i, j, $\mathrm{val}(f_j^{(i)}) \geq \mathrm{val}(f_j) - i$. The result follows. □

The previous lemma is combined with a bound on the valuation of a specific Wronskian.

Lemma 2.5. *Let $f_j = X^{\alpha_j}(uX+v)^{\beta_j}$, $1 \leq j \leq \ell$, such that $\alpha_j, \beta_j \geq \ell$ for all j and $uv \neq 0$. If the f_j's are linearly independent, then*

$$\mathrm{val}(\mathrm{wr}(f_1, \dots, f_\ell)) \leq \sum_{j=1}^{\ell} \alpha_j.$$

Proof. By Leibniz rule, for all i, j

$$f_j^{(i)}(X) = \sum_{t=0}^{i} \binom{i}{t} (\alpha_j)_t (\beta_j)_{i-t} u^{i-t} X^{\alpha_j - t} (uX+v)^{\beta_j - i + t}$$

where $(m)_n = m(m-1)\cdots(m-n+1)$ is the falling factorial. Since $\alpha_j - t \geq \alpha_j - i$ and $\beta_j - i + t \geq \beta_j - i$ for all t,

$$f_j^{(i)}(X) = X^{\alpha_j - i}(uX+v)^{\beta_j - i} \times \sum_{t=0}^{i} \binom{i}{t} (\alpha_j)_t (\beta_j)_{i-t} u^{i-t} X^{i-t} (uX+v)^t.$$

Furthermore, since $\alpha_j \geq \ell \geq i$, we can write $X^{\alpha_j - i} = X^{\alpha_j - \ell} X^{\ell - i}$ and since $\beta_j \geq \ell \geq i$, $(uX+v)^{\beta_j - i} = (uX+v)^{\beta_j - \ell}(uX+v)^{\ell - i}$. The entries of the Wronskian matrix W of f_1, $\dots$, f_ℓ are $W_{ij} = f^{(i)_j}$ for $0 \leq i < \ell$ and $1 \leq j \leq \ell$. Thus, the entries of the column j of W share the common factor $X^{\alpha_j - \ell}(uX+v)^{\beta_j - \ell}$, and the entries of the row i share the common factor $X^{\ell - i}(uX+v)^{\ell - i}$. Together, we get

$$\mathrm{wr}(f_1, \dots, f_\ell) = X^{\sum_j \alpha_j - \binom{\ell}{2}} (uX+v)^{\sum_j \beta_j - \binom{\ell}{2}} \det(M)$$

where the matrix M is defined by

$$M_{i,j} = \sum_{t=0}^{i} \binom{i}{t} (\alpha_j)_t (\beta_j)_{i-t} u^{i-t} X^{i-t} (uX+v)^t.$$

The polynomial $\det(M)$ is nonzero since the f_j's are supposed linearly independent and its degree is at most $\binom{\ell}{2}$. Therefore its valuation cannot be larger than its degree and is bounded by $\binom{\ell}{2}$.

Altogether, the valuation of the Wronskian is bounded by $\sum_j \alpha_j - \binom{\ell}{2} + \binom{\ell}{2} = \sum_j \alpha_j$. □

Proof of Theorem 2.1. Let $P = \sum_j a_j X^{\alpha_j}(uX+v)^{\beta_j}$, and let $f_j = X^{\alpha_j}(uX+v)^{\beta_j}$. We assume first that $\alpha_j, \beta_j \geq \ell$ for all j, and that the f_j's are linearly independent. Note that $\mathrm{val}(f_j) = \alpha_j$ for all j.

Let W denote the Wronskian of the f_j's. We can replace f_1 by P in the first column of the Wronskian matrix using column operations which multiply the determinant by a_1 (its valuation does not change). The matrix we obtain is the Wronskian matrix of $P, f_2, \dots, f_\ell$. Now using Lemma 2.4, we get

$$\mathrm{val}(W) \geq \mathrm{val}(P) + \sum_{j\geq 2} \alpha_j - \binom{\ell}{2}.$$

This inequality combined with Lemma 2.5 shows that

$$\mathrm{val}(P) \leq \alpha_1 + \binom{\ell}{2}. \tag{1}$$

We now aim to remove our two previous assumptions. If the f_j's are not linearly independent, we can extract from this family a basis $f_{j_1}, \dots, f_{j_d}$. Then P can be expressed in this basis as $P = \sum_{l=1}^{d} \tilde{a}_l f_{j_l}$. We can apply Equation (1) to $f_{j_1}, \dots, f_{j_d}$ and obtain $\mathrm{val}(P) \leq \alpha_{j_1} + \binom{d}{2}$. Since $j_d \leq \ell$, we have $j_1 + d - 1 \leq \ell$ and $\mathrm{val}(P) \leq \alpha_{j_1} + \binom{\ell+1-j_1}{2}$. The value of j_1 being unknown, we conclude that

$$\mathrm{val}(P) \leq \max_{1\leq j\leq \ell}\left(\alpha_j + \binom{\ell+1-j}{2}\right). \tag{2}$$

The second assumption is that $\alpha_j, \beta_j \geq \ell$. Given P, consider $\tilde{P} = X^\ell (uX+v)^\ell P = \sum_j a_j X^{\tilde{\alpha}_j}(uX+v)^{\tilde{\beta}_j}$. Then $\tilde{P}$ satisfies $\tilde{\alpha}_j, \tilde{\beta}_j \geq \ell$, whence by Equation (2), $\mathrm{val}(\tilde{P}) \leq \max_j(\tilde{\alpha}_j + \binom{\ell+1-j}{2})$. Since $\mathrm{val}(\tilde{P}) = \mathrm{val}(P) + \ell$ and $\tilde{\alpha}_j = \alpha_j + \ell$, the result follows. $\square$

2.2 Generalization

We first state a generalization of Theorem 2.1 to a sum of product of powers of low-degree polynomials. Then we state a special case of this generalization that is useful for computing multilinear factors.

Theorem 2.6. *Let $(\alpha_{i,j}) \in \mathbb{Z}_+^{m\times \ell}$ and*

$$P = \sum_{j=1}^{\ell} a_j \prod_{i=1}^{m} f_i^{\alpha_{i,j}} \in \mathbb{K}[X],$$

where the degree of $f_i \in \mathbb{K}[X]$ is d_i for all i. Let $F \in \mathbb{K}[X]$ be an irreducible polynomial and let $\mu_i = \mu_F(f_i)$ for all i. Then the multiplicity $\mu_F(P)$ of F as a factor of P satisfies

$$\mu_F(P) \leq \max_{1\leq j\leq \ell} \sum_{i=1}^{m}\left(\mu_i \alpha_{i,j} + (d_i - \mu_i)\binom{\ell+1-j}{2}\right).$$

Proof. Let $P_j = \prod_{i=1}^m f_i^{\alpha_{i,j}}$ for $1 \leq j \leq \ell$. As in the proof of Theorem 2.1, we can assume without loss of generality that the P_j's are linearly independent, and the $\alpha_{i,j}$'s not less than ℓ.

We can use a generalized Leibniz rule to compute the derivatives of the P_j's. Namely

$$P_j^{(T)} = \sum_{t_1+\cdots+t_m=T} \binom{T}{t_1,\ldots,t_m} \prod_{i=1}^m (f_i^{\alpha_{i,j}})^{(t_i)}, \tag{3}$$

where $\binom{T}{t_1,\ldots,t_m} = T!/(t_1!\cdots t_m!)$ is the multinomial coefficient. Consider now a derivative of the form $(f^\alpha)^{(t)}$. This is a sum of terms, each of which contains a factor $f^{\alpha-t}$. In Equation (3), each t_i is bounded by T. This means that $P_j^{(T)} = Q_{T,j}\prod_i f_i^{\alpha_{i,j}-T}$ for some polynomial $Q_{T,j}$. Since the degree of $P_j^{(T)}$ equals $\sum_i d_i\alpha_{i,j} - T$, $Q_{T,j}$ has degree $\sum_i d_i\alpha_{i,j} - T - \sum_i(d_i\alpha_{i,j} - d_iT) = (\sum_i d_i - 1)T$.

Consider now $W = \mathrm{wr}(P_1,\ldots,P_\ell)$. We can factor out $\prod_i f_i^{\alpha_{i,j}-\ell}$ in each column and $\prod_i f_i^{\ell-T}$ in each row. At row T and column j, we therefore factor out $\prod_i f_i^{\alpha_{i,j}-\ell} \cdot \prod_i f_i^{\ell-T} = \prod_i f_i^{\alpha_{i,j}-T}$. Thus,

$$W = \prod_{i=1}^m f_i^{\sum_j \alpha_{i,j} - \binom{\ell}{2}} \det(M)$$

where $M_{T,j} = Q_{T,j}$. Thus, $\det(M)$ is a polynomial of degree at most $(\sum_i d_i - 1)\binom{\ell}{2}$.

Therefore, the multiplicity $\mu_F(W)$ of F as a factor of W is bounded by its multiplicity as a factor of $\prod_i f_i^{\sum_j \alpha_{i,j} - \binom{\ell}{2}}$ plus the degree of $\det(M)$. We get

$$\begin{aligned} \mu_F(W) &\leq \sum_i \mu_i \left(\sum_j \alpha_{i,j} - \binom{\ell}{2}\right) + (\sum_i d_i - 1)\binom{\ell}{2} \\ &= \sum_i \left(\mu_i \sum_j \alpha_{i,j} + (d_i - \mu_i)\binom{\ell}{2}\right) - \binom{\ell}{2}. \end{aligned} \tag{4}$$

To conclude the proof, it remains to remember Lemma 2.4 and use the same proof technique as in Theorem 2.1. It was expressed in terms of the valuation of the polynomials, but remains valid with the multiplicity of any factor. In this case, it can be written as $\mu_F(W) \geq \sum_j \mu_F(P_j) - \binom{\ell}{2}$ where W is the Wronskian of the P_j's. Using column operations, we can replace the first column of the Wronskian matrix of the P_j's by the polynomial P and its derivatives. We get $\mu_F(W) \geq \mu_F(P) + \sum_{j\geq 2} \mu_F(P_j) - \binom{\ell}{2}$, where $\mu_F(P_j) = \sum_i \mu_i\alpha_{i,j}$.

Together with (4), we get

$$\begin{aligned}\mu_F(P) &\leq \mu_F(W) - \sum_{j\geq 2}\mu_F(P_j) + \binom{\ell}{2}\\ &\leq \sum_i\left(\mu_i\sum_j \alpha_{i,j} + (d_i-\mu_i)\binom{\ell}{2}\right) - \binom{\ell}{2} - \sum_{j\geq 2}\sum_i \mu_i\alpha_{i,j} + \binom{\ell}{2}\\ &\leq \sum_i\left(\mu_i\alpha_{i1} + (d_i-\mu_i)\binom{\ell}{2}\right).\end{aligned}$$

To obtain the bound of the theorem, the two initial assumption have to be removed using the same technique as in Theorem 2.1. □

In the previous proof, the multiplicity of F as a factor of $\det(M)$ is bounded by the degree of $\det(M)$. One can actually bound this multiplicity by the degree of $\det(M)$ divided by the degree of F. If we denote by d_F the degree of F, the bound of the theorem can be refined to

$$\mu_F(P) \leq \max_{1\leq j\leq \ell}\left[\sum_{i=1}^{m}\left(\mu_i\alpha_{i,j} + \left(\frac{d_i}{d_F}-\mu_i\right)\binom{\ell+1-j}{2}\right) + \left(1-\frac{1}{d_F}\right)\binom{\ell+1-j}{2}\right].$$

As a special case of the theorem, one obtains the following corollary.

Corollary 2.7. *Let $P = \sum_{j=1}^{\ell} a_j X^{\alpha_j}(vX+t)^{\beta_j}(uX+w)^{\gamma_j}$, $wt \neq 0$. If P is nonzero then its valuation is at most $\max_{1\leq j\leq \ell}(\alpha_j + 2\binom{\ell+1-j}{2})$.*

2.3 Is Theorem 2.1 tight?

Let P be as in Theorem 2.1, that is

$$P = \sum_{j=1}^{\ell} a_j X^{\alpha_j}(uX+v)^{\beta_j}$$

where $uv \neq 0$. If the family $(X^{\alpha_j}(1+X)^{\beta_j})_{1\leq j\leq \ell}$ is linearly independent over $\mathbb{K}$, the valuation of P is at most $\alpha_1 + \binom{\ell}{2}$. In this section, we investigate the tightness of this bound. In the special case $\alpha_j = \alpha_1$ for all j, Hajós' Lemma [21, 37] gives the better bound $\alpha_1 + (\ell - 1)$. (This bound can be shown to be tight by expanding $X^{\ell-1} = (-1+(X+1))^{\ell-1}$ with the binomial formula.)

We first prove that Hajós bound does not apply when the α_j's are not all equal by constructing a family of polynomials of valuation $2\ell-3$ where ℓ is the number of terms. Note that in the following proposition, the polynomial has $\ell' = \ell+3$ terms and valuation $2\ell'-3 = 2\ell+3$.

Proposition 2.8. *For $\ell \geq 0$,*

$$X^{2\ell+3} = -1 + (1+X)^{2\ell+3} + \sum_{j=0}^{\ell} \frac{2\ell+3}{2j+1}\binom{\ell+1+j}{\ell+1-j} X^{2j+1}(1+X)^{\ell+1-j}.$$

Furthermore, the family $(1, (1+X)^{2\ell+3}, X(1+X)^{\ell+1}, X^3(1+X)^{\ell}, \ldots, X^{2\ell+1}(1+X))$ *is linearly independent.*

Proof. It is clear that the polynomial has degree $(2\ell+3)$ and is monic. Furthermore, the family is linearly independent since the polynomials have pairwise different degrees.

Let P_ℓ be the right-hand side polynomial in the proposition and $[X^m]P_\ell$ be the coefficient of the monomial X^m in P_ℓ. We shall prove that $[X^m]P_\ell = 0$ for all $m < 2\ell+3$. It is clear for $m = 0$. For $m > 0$,

$$[X^m]P_\ell = \binom{2\ell+3}{m} - \sum_{j=0}^{\ell} \frac{2\ell+3}{2j+1}\binom{\ell+1+j}{\ell+1-j}\binom{\ell+1-j}{m-2j-1}. \tag{5}$$

We prove that $[X^m]P_\ell = 0$ with the help of the implementation of Wilf and Zeilberger's algorithm in the Maple package EKHAD of Doron Zeilberger [38]. For $0 \leq j \leq \ell$, let $F(m, j)$ be the summand in equation (5) divided by $\binom{2\ell+3}{m}$. The package EKHAD provides the recurrence

$$mF(m+1, j) - mF(m, j) = F(m, j+1)R(m, j+1) - F(m, j)R(m, j)$$

where

$$R(m, j) = \frac{2j(2j+1)(\ell+j+2-m)}{(2\ell+3-m)(2j-m)}.$$

To conclude the proof, it is sufficient to prove the recurrence relation and that it implies the equality. These are simple computations that we omit but can easily be checked by hand. That is, the validity of the proof does not rely in any manner on the validity of the computer program that was used to produce it. ☐

We now address the question of improving the bound in Theorem 2.1. The bound is obtained as a combination of a lower bound, given by Lemma 2.4, and an upper bound, given by Lemma 2.5. None of the bounds can be improved in the general case: The lower bound is tight if all the valuations are distinct while the upper bound is tight if all the valuations are equal. On the other hand, both bounds can be improved in some special cases. Let $f_1, \ldots, f_\ell \in \mathbb{K}[X]$ ordered by increasing valuations, such that $\mathrm{val}(f_j) \leq \mathrm{val}(f_1) + (j-1)$ for all $j > 0$. Then $\mathrm{val}(\mathrm{wr}(f_1, \ldots, f_\ell)) \geq \ell\,\mathrm{val}(f_1)$ since for all j, f_j can be replaced in the Wronskian matrix by some g_j of valuation at least $\mathrm{val}(f_1)+(j-1)$ (proof omitted). And if on the contrary the valuations are all distinct, the matrix made of the constant coefficients of the entries of the matrix M defined in the proof of Lemma 2.5 is similar to a Vandermonde matrix. This implies that the valuation of the Wronskian equals $\sum_j \alpha_j - \binom{\ell}{2}$. These two improvements are enough to recover Hajós' bound on the one hand, and the obvious fact that $\operatorname{val}(P) = \alpha_1$ if the α_j's are pairwise distinct on the other hand. Unfortunately, they are not sufficient to improve Theorem 2.1 in the general case. For instance, if $\operatorname{val}(f_j) = \operatorname{val}(f_1) + \binom{j-1}{2}$ for all j, we are not able to improve any of the two bounds.

3 Gap Theorems and their application to PIT

In this section, we first prove our Gap Theorems. Then we give very direct applications of these theorems in the form of Polynomial Identity Testing algorithms for some families of univariate polynomials.

3.1 Two Gap Theorems

We still assume that the coefficients of the polynomials we consider lie in some field $\mathbb{K}$ of characteristic zero.

The bound on the valuation obtained in the Section 2.1 translates into a Gap Theorem for linear factors.

Theorem 3.1 (Gap Theorem for linear factors). *Let $P = Q + R$ where*

$$Q = \sum_{j=1}^{\ell} a_j X^{\alpha_j} Y^{\beta_j} \text{ and } R = \sum_{j=\ell+1}^{k} a_j X^{\alpha_j} Y^{\beta_j}$$

such that $\alpha_1 \le \cdots \le \alpha_k$. Suppose that ℓ is the smallest index such that $\alpha_{\ell+1} > \alpha_1 + \binom{\ell}{2}$. Then, for every $F = uX + vY + w$ with $uvw \neq 0$,

$$\mu_F(P) = \min(\mu_F(Q), \mu_F(R)).$$

Proof. Let $F = uX + vY + w$, $uvw \neq 0$. Then F divides P if and only if $P(X, -\frac{1}{v}(uX + w)) = 0$, and the same holds for the polynomials Q and R. Let $P^\star(X) = P(X, -\frac{1}{v}(uX + w))$, and define $Q^\star$ and $R^\star$ in the same way from Q and R.

Let us first prove that F divides P if and only if it divides both Q and R. Clearly, F divides P if it divides both Q and R. Suppose that F does not divide Q, that is $Q^\star$ is nonzero. By Theorem 2.1, its valuation is at most $\max_{j\le\ell}(\alpha_j + \binom{\ell+1-j}{2})$. Furthermore,

$\alpha_{j+1} \le \alpha_1 + \binom{j}{2}$ for all $j < \ell$ by hypothesis. Therefore,

$$\begin{aligned} \mathrm{val}(Q^\star) &\le \max_{1\le j\le \ell}\left(\alpha_j + \binom{\ell+1-j}{2}\right) \\ &\le \max_{1\le j\le \ell}\left(\alpha_1 + \binom{j-1}{2} + \binom{\ell+1-j}{2}\right) \\ &\le \alpha_1 + \binom{\ell}{2}. \end{aligned}$$

The last inequality holds since $\binom{j-1}{2} + \binom{\ell-(j-1)}{2} \le \binom{\ell}{2}$ for $1 \le j \le \ell$.

The valuation of $R^\star$ is at least $\alpha_{\ell+1}$ which is by hypothesis larger than $\alpha_1 + \binom{\ell}{2}$. Therefore, if $Q^\star$ is not identically zero, its monomial of lowest degree cannot be canceled by a monomial of $R^\star$. In other words, $P^\star = Q^\star + R^\star$ is nonzero and F does not divide P. Finally, if F divides Q but not R, it cannot divide P since it would divide $P - Q = R$.

To show that $\mu_F(P) = \min(\mu_F(Q), \mu_F(P))$, we remark that F is a factor of multiplicity μ of P if and only if it divides $\partial^m P/\partial X^m$ for all $m \le \mu$. Since $\mu = \mu_F(P) = \mu_F(X^d P)$ for all d, one can assume that $\alpha_1 \ge \mu$. Then

$$\frac{\partial^m P}{\partial X^m} = \sum_{j=1}^{k} a_j (\alpha_j)_m X^{\alpha_j - m} Y^{\beta_j}$$

for $1 \le m \le \mu$. The hypothesis $\alpha_{\ell+1} > \alpha_1 + \binom{\ell}{2}$ in the theorem only depends on the difference between the exponents. By linearity of the derivative, the previous argument actually shows that F divides $\partial^m P/\partial X^m$ if and only if it divides both $\partial^m Q/\partial X^m$ and $\partial^m R/\partial X^m$. This proves that $\mu_F(P) = \min(\mu_F(Q), \mu_F(R))$. □

It is straightforward to extend this theorem to more *gaps*. The theorem can be recursively applied to Q and R (as defined in the proof). Then, if $P = P_1 + \ldots + P_s$ where there is a *gap* between P_t and P_{t+1} for $1 \le t < s$, then any linear polynomial $(uX + vY + w)$ is a factor of multiplicity μ of P if and only if it is a factor of multiplicity at least μ of each P_t. Moreover, one can write every P_t as $X^{q_t} Q_t$ such that X does not divide Q_t. Then the degree in X of Q_t is bounded by $\binom{\ell_t - 1}{2}$ where ℓ_t is its number of terms.

The following definition makes this discussion formal.

Definition 3.2. Let P as in Theorem 3.1. A set $\mathscr{Q} = \{Q_1, \ldots, Q_s\}$ is a *decomposition of P with respect to X* if there exist integers $q_1, \ldots, q_s$ such that $P = X^{q_1} Q_1 + \cdots + X^{q_s} Q_s$ with $q_t > q_{t-1} + \deg_X(Q_{t-1})$ for $1 < t \le s$.

A decomposition is *compatible* with a set $\mathscr{F}$ of polynomials if for all $F \in \mathscr{F}$, $\mu_F(P) = \min_{1\le t\le s} \mu_F(Q_t)$.

The *degree* in X (resp. in Y) of a decomposition is the sum of the degrees in X (resp. in Y) of the Q_t's.

The Gap Theorem implies a decomposition of P of degree at most $\binom{k-1}{2}$ in X. Indeed, the degree of each Q_t is at most $\binom{\ell_t-1}{2}$, with $\sum_t \ell_t = k$, and the function $k \mapsto \binom{k}{2}$ is super-additive, *i.e.*, $\binom{i+j}{2} \geq \binom{i}{2} + \binom{j}{2}$.

Remark 3.3. Let $\mathscr{Q}'$ be the decomposition obtained from $\mathscr{Q}$ by replacing Q_t and Q_{t+1} by $Q'_t = Q_t + X^{q_{t+1}-q_t} Q_{t+1}$. It is easy to see that if $\mathscr{Q}$ is compatible with a set $\mathscr{F}$, then so is $\mathscr{Q}'$. More generally, one obtains compatible decompositions by grouping together any number of consecutive polynomials in a decomposition.

Using the generalization of Theorem 2.1 given in Section 2.2, one can also prove a similar Gap Theorem, but for multilinear factors.

Theorem 3.4 (Gap Theorem for multilinear factors). *Let $P = Q + R$ where*

$$Q = \sum_{j=1}^{\ell} a_j X^{\alpha_j} Y^{\beta_j} \text{ and } R = \sum_{j=\ell+1}^{k} a_j X^{\alpha_j} Y^{\beta_j},$$

such that $\alpha_1 \leq \cdots \leq \alpha_k$. If ℓ is the smallest index such that $\alpha_{\ell+1} > \alpha_1 + 2\binom{\ell}{2}$ then for every $F = uXY + vX + wY + t$ with $wt \neq 0$,

$$\mu_F(P) = \min(\mu_F(Q), \mu_F(R)).$$

Proof. Let $F = uXY + vX + wY + t$, $wt \neq 0$. Then F divides P if and only if $P(X, -\frac{vX+t}{uX+w}) = 0$. This is equivalent to the fact that the polynomial

$$(uX + w)^B P(X, -\frac{vX + t}{uX + w}) = \sum_{j=1}^{k} a_j X^{\alpha_j} (-vX - t)^{\beta_j} (uX + w)^{B-\beta_j}$$

vanishes, where $B = \max_j \beta_j$. The rest of the proof is identical to the proof of the Gap Theorem for linear factors, using the valuation bound of Corollary 2.7 instead of Theorem 2.1. $\square$

3.2 Polynomial Identity Testing

The first algorithmic applications of our technical results are two polynomial identity testing algorithms. The problem is the following: Given a polynomial in some specified representation, decide whether it is identically zero. Of course, if the polynomial is given as the list of its coefficient, the problem is trivial. In the most general forms of the problem, the polynomial is represented either as an arithmetic circuit or a blackbox. Then, it is not clear how to perform the test since expanding the polynomial as a sum of monomial may require exponential time. We refer the reader to the survey of Shpilka and Yehudayoff [40] for more details and recent results on this problem.

Our algorithms deal with univariate high-degree polynomials, represented as sums of product of powers of dense polynomials. In particular, one can represent such polynomials by small arithmetic circuits. If the polynomial is known to be sparse, one can apply the algorithm of Bläser et al. [4]. Koiran et al. [32] and Grenet et al. [19] give results close to the ones we prove here. A comparison is given at the end of this section.

The polynomials in this section have coefficients in an algebraic number field $\mathbb{K} = \mathbb{Q}[\xi]/\langle\varphi\rangle$ where $\varphi \in \mathbb{Q}[\xi]$ is irreducible. An element e of $\mathbb{K}$ is uniquely represented by a polynomial $p_e \in \mathbb{Q}[\xi]$ of degree smaller than $\deg(\varphi)$. In the algorithms, a coefficient $c \in \mathbb{K}$ of a lacunary polynomial is given as the dense representation of p_c, that is the list of all its coefficients including the zero ones. Moreover, the algorithms are uniform in $\mathbb{K}$ in the sense that they can take as input the polynomial φ defining $\mathbb{K}$.

Theorem 3.5. *Let $\mathbb{K}$ be an algebraic number field and*

$$P = \sum_{j=1}^{k} a_j X^{\alpha_j}(uX+v)^{\beta_j} \in \mathbb{K}[X].$$

There exists a deterministic polynomial-time algorithm to decide if P vanishes.

Proof. We assume without loss of generality that $\alpha_{j+1} \geq \alpha_j$ for all j and $\alpha_1 = 0$. If α_1 is nonzero, X^{α_1} divides P and we consider P/X^{α_1}.

Suppose first that $u = 0$. Then P is given as a sum of monomials, and we only have to test each coefficient for zero. Note that the α_j's are not necessarily distinct. Thus the coefficients are of the form $\sum_j a_j v^{\beta_j}$. Lenstra [35] gives an algorithm to find low-degree factors of univariate lacunary polynomials. It is easy to deduce from his algorithm an algorithm to test such sums for zero. A strategy could be to simply apply Lenstra's algorithm to $\sum_j a_j X^{\beta_j}$ and then check whether $(X - v)$ is a factor, but one can actually improve the complexity by extracting from his algorithm the relevant part (we omit the details). The case $v = 0$ is similar.

We assume now that $uv \neq 0$. Then $P = 0$ if and only if $(Y - uX - v)$ divides $\sum_j a_j X^{\alpha_j} Y^{\beta_j}$. Recursively using the Gap Theorem for linear factors (Theorem 3.1), one computes a decomposition $P = X^{q_1}Q_1 + \cdots + X^{q_s}Q_s$ such that P is identically zero if and only if each Q_t in this sum is also. Therefore, we are left with testing if each Q_t is identically zero.

To this end, let Q be one these polynomials. With a slight abuse of notation, it can be written $Q = \sum_{j=1}^{k} a_j X^{\alpha_j}(uX+v)^{\beta_j}$. It satisfies $\alpha_1 = 0$ and $\alpha_{j+1} \leq \binom{j}{2}$ for all j.

In particular, $\alpha_k \leq \binom{k-1}{2}$. Consider the change of variables $Y = uX + v$. Then

$$Q(Y) = \sum_{j=1}^{k} a_j u^{-\alpha_j} (Y - v)^{\alpha_j} Y^{\beta_j}$$

is identically zero if and only if $Q(X)$ is. We can express $Q(Y)$ as a sum of powers of Y:

$$Q(Y) = \sum_{j=1}^{k} \sum_{\ell=0}^{\alpha_j} a_j u^{-\alpha_j} \binom{\alpha_j}{\ell} (-v)^{\ell} Y^{\alpha_j + \beta_j - l}.$$

There are at most $k\binom{k-1}{2} = \mathcal{O}(k^3)$ monomials. Then, testing if $Q(Y)$ is identically zero consists in testing whether each coefficient vanishes. Moreover, each coefficient has the form $\sum_j \binom{\alpha_j}{\ell_j} a_j u^{-\alpha_j} (-v)^{\ell_j}$ where the sum ranges over at most k indices. Since $\ell_j, \alpha_j \leq \binom{k-1}{2}$ for all j, the terms in these sums have polynomial bit-lengths. Therefore, the coefficients can be tested for zero in polynomial time.

Altogether, this gives a polynomial-time algorithm to test if P is identically zero. $\square$

One can actually replace the linear polynomial $(uX + v)$ in the previous theorem by any binomial polynomial. Without loss of generality, one can consider that this binomial is $(uX^d + v)$, and we assume that it is represented in lacunary representation. In other words, its size is polynomial in $\log(d)$.

Corollary 3.6. *Let $\mathbb{K}$ be an algebraic number field and*

$$P = \sum_{j=1}^{k} a_j X^{\alpha_j} (uX^d + v)^{\beta_j} \in \mathbb{K}[X].$$

There exists a deterministic polynomial-time algorithm to decide if the polynomial P vanishes.

Proof. For all j we consider the Euclidean division of α_j by d: $\alpha_j = q_j d + r_j$ with $r_j < d$. We rewrite P as

$$P = \sum_{j=1}^{k} a_j X^{r_j} (X^d)^{q_j} (uX^d + v)^{\beta_j}.$$

Let us group in the sum all the terms with a common r_j. That is, let

$$P_i(Y) = \sum_{\substack{1 \leq j \leq k \\ r_j = i}} a_j Y^{q_j} (uY + v)^{\beta_j}$$

for $0 \leq i < d$. We remark that regardless of the value of d, the number of nonzero P_i's is bounded by k. We have $P(X) = \sum_{i=0}^{d-1} X^i P_i(X^d)$. Each monomial X^α of $X^i P_i(X^d)$ satisfies $\alpha \equiv i \mod d$. Therefore, P is identically zero if and only if all the P_i's are identically zero.

Since each P_i has the same form as in Theorem 3.5, and there are at most k of them, the previous algorithm can be applied to each of them to decide the nullity of P. □

Using Theorem 2.6 instead of Theorem 2.1, one can give a polynomial identity testing algorithm for a larger class of polynomials. In the next algorithm, we use the *dense representation* for univariate polynomials, that is the list of all their coefficients (including the zero ones). Then *dense size* of a univariate polynomial is then the sum of the sizes of its coefficients.

Theorem 3.7. *Let $f_1, \ldots, f_m$ be monic univariate polynomials over a number field. Let*

$$P = \sum_{j=1}^{k} a_j \prod_{i=1}^{m} f_i^{\alpha_{i,j}}.$$

There is a deterministic algorithm to decide if P is zero whose running time is polynomial in k, m, the dense sizes of the f_i's and the bitsizes of the a_j's and the $\alpha_{i,j}$'s.

With an oracle to decide the nullity of sums of the form $\sum_j \prod_i \lambda_i^{\alpha_{i,j}}$ where the λ_i's are in the number field, the above algorithm can be used with non monic polynomials $f_1, \ldots, f_m$.

Proof. Let us suppose that the f_i's are not monic. In time polynomial in the degree of the f_i's, one can compute their monic irreducible factorizations. Then we can write $f_i = \lambda_i \prod_{t=1}^{n} g_t^{\beta_{i,t}}$ for all i, where the g_t's are distinct monic irreducible polynomials and $\beta_{i,t} \geq 0$. Then, the polynomial P can be written as

$$P = \sum_{j=1}^{k} \left[a_j \left(\prod_{i=1}^{m} \lambda_i^{\alpha_{i,j}} \right) \left(\prod_{t=1}^{n} g_t^{\sum_i \beta_{i,t}\alpha_{i,j}} \right) \right].$$

For all j and t, let $\gamma_{j,t} = \sum_i \beta_{i,t}\alpha_{i,j}$ and $\Lambda_j = \prod_i \lambda_i^{\alpha_{i,j}}$. Note that the case of monic polynomials is the case where $\Lambda_j = 1$.

If $n = 1$, that is all the f_i's are powers of a same polynomial g, then $P = \sum_j a_j \Lambda_j g^{\gamma_j}$. It is thus sufficient to find the subsets of indices for which γ_j is constant, and test for zero sums of the form $\sum_j a_j \Lambda_j$. These sums can be easily tested for zero if $\Lambda_j = 1$, and using the oracle otherwise.

If $n > 1$, we use a Gap Theorem. To this end, we use the bound on the multiplicity of the irreducible polynomial g_1 given by Theorem 2.6. Let

$$Q = \sum_{j=1}^{\ell} a_j \Lambda_j \prod_{t=1}^{n} g_t^{\gamma_{j,t}} \text{ and } R = \sum_{j=\ell+1}^{k} a_j \Lambda_j \prod_{t=1}^{n} g_t^{\gamma_{j,t}}$$

and suppose that ℓ is the smallest index such that

$$\gamma_{\ell+1,1} > \gamma_{1,1} + \Big(\sum_{t>1} \deg(g_t)\Big)\binom{\ell}{2}. \tag{6}$$

Since $\mu_{g_1}(g_t) = 0$ for all $t > 1$ and $\mu_{g_1}(g_1) = 1$, Theorem 2.6 implies that

$$\begin{aligned}\mu_{g_1}(Q) &\le \max_{j\le\ell}\left(\gamma_{1,j} + \sum_{t>1} \deg(g_t)\binom{\ell-j+1}{2}\right)\\ &\le \gamma_{1,1} + \Big(\sum_{t>1} \deg(g_t)\Big)\binom{\ell}{2}.\end{aligned}$$

The second inequality can be proved exactly as in the proof of the Gap Theorem for linear factors. In particular, one has $\gamma_{\ell+1,1} > \mu_{g_1}(Q)$. Therefore, if Q is nonzero, since $\mu_{g_1}(R) \ge \gamma_{\ell+1,1}$, then $P = Q + R$ is nonzero. The same argument of course works for any g_t.

Algorithmically, we can decompose P into $Q + R$ using Equation (6), and recursively decompose R. In each polynomial of the decomposition we factor out the largest possible power of g_1. We obtain $P = g_1^{\gamma(1)} Q_1 + \cdots + g_1^{\gamma(p)} Q_p$ for some p such that $P = 0$ if and only if each $Q_1 = \cdots = Q_p = 0$. Furthermore, the exponents of g_1 in $Q_1, \ldots, Q_p$ are bounded by $\binom{k}{2}\sum_{t>1}\deg(g_t)$. Let us call $\{Q_1, \ldots, Q_p\}$ the decomposition of P with respect to g_1.

The algorithm is then as follows. We initially consider the set $\mathcal{Q} = \{P\}$. Then for $t = 1$ to n, we replace each member of $\mathcal{Q}$ by its decomposition with respect to g_t. We obtain a set $\mathcal{Q}$ of polynomials such that $P = 0$ if and only if $Q = 0$ for all $Q \in \mathcal{Q}$.

It remains to test whether each polynomial of $\mathcal{Q}$ vanishes. To this end, one can expand these polynomials as sums of monomials since their degrees are polynomially bounded. The coefficients of the monomials will have the form $\sum_j a_j \Lambda_j c_{j,\delta}$ where $c_{j,\delta}$ is the coefficient of X^δ in the polynomial $\prod_t g_t^{\gamma_{j,t}}$. Therefore, it has a polynomial bitsize. Testing the nullity of these coefficients is then easy if $\Lambda_j = 1$ and done using the oracle otherwise.

This proves the theorem. $\square$

Note that an alternative algorithm for the same class of polynomials is given by Koiran et al. [32], in the case where the f_i's are not monic. Yet the complexity of their algorithm is exponential in k and m. We also remark that our algorithm uses, for non monic polynomials, an oracle to decide the nullity of a sum of products of powers of integers (or elements of a number field). The same oracle is also needed in the polynomial identity testing algorithm of Grenet et al. [19]. This algorithm deals with a very similar family of polynomial, where the f_i's are sparse polynomials instead of dense ones, but the complexity is then polynomial in the exponents α_{ij}'s rather than in their bitsizes.

4 Factoring lacunary polynomials

We now turn to the main applications of our Gap Theorems of Section 3. We detail how to compute linear and multilinear factors of lacunary polynomials over number fields. We first focus on bivariate polynomials. In Section 4.3, we explain that our algorithms can be easily extended to multivariate polynomials. We discuss the case of other fields of characteristic zero in Section 4.4.

The input polynomials in our algorithms are given in lacunary representation. A monomial is represented by its coefficient and its vector of exponents written in binary. For the representation of elements of a number field, we refer to the discussion opening Section 3.2. The size of the lacunary representation is the sum of the sizes of the representations of the nonzero monomials. In particular, note that the size is polynomial in the logarithm of the degree when the number of terms and the coefficient heights are fixed.

As we shall see, finding binomial factors is a special case in our algorithms. To simplify the next proofs, we first prove a lemma on the computation of these factors.

Lemma 4.1. *Let $\mathbb{K}$ be an algebraic number field and*

$$P = \sum_{j=1}^{k} a_j X^{\alpha_j} Y^{\beta_j} \in \mathbb{K}[X, Y].$$

There exists a deterministic polynomial-time algorithm that computes all the multilinear binomial factors of P, together with their multiplicities.

Proof. Let F be an irreducible binomial factor of P. Without loss of generality, let us assume that it depends on X. Otherwise, invert the roles of X and Y in what follows. Then F can be written $F = uXY^\gamma + vY^\delta$ with $u, v \in \mathbb{K}$ and $\gamma + \delta \leq 1$. Then F divides P if and only if $P(-\frac{v}{u}Y^{\delta-\gamma}, Y) = 0$. In other words, F divides P if and only if $G = uZ + v$ is a factor of the polynomial $Q(Y, Z) = Y^{\max_j \alpha_j} P(ZY^{\delta-\gamma}, Y)$. One can view Q as an element of $\mathbb{K}[Z][Y]$ and write it as $Q = \sum_\epsilon q_\epsilon(Z) Y^\epsilon$. Then G divides Q if and only if it divides each q_ϵ. More precisely, the multiplicity $\mu_F(P)$ of F as a factor of P equals $\min_\epsilon \mu_G(q_\epsilon)$. Therefore, it is sufficient to compute the linear factors of each q_ϵ, with multiplicities, using Lenstra's algorithm [35].

To find all the multilinear binomial factors of P depending on X, one has to apply the above algorithm with $(\gamma,\delta)=(1,0)$, $(0,1)$ and $(0,0)$. For the factors depending only on Y, one has to invert the roles of X and Y and apply to above algorithm once more. $\square$

Note that in the previous algorithm, it is not efficient to first compute the gcd of the polynomials q_ϵ before the computation of their common linear factors since this task is NP-hard to perform [39].

4.1 Finding linear factors

Theorem 4.2. *Let $\mathbb{K}$ be a number field and*

$$P(X,Y)=\sum_{j=1}^{k} a_j X^{\alpha_j} Y^{\beta_j} \in \mathbb{K}[X,Y].$$

Then there exists a deterministic polynomial-time algorithm that finds all the linear factors of P, together with their multiplicities.

Proof. The algorithm has three distinct parts. The first part is the obvious computation of monomial factors, the second part is for binomial factors using Lemma 4.1, and the third part for trinomial factors.

Consider the factors of the form $F=uX+vY+w$, $uvw\neq 0$. Using recursively the Gap Theorem for linear factors (Theorem 3.1), we can compute a decomposition of P with respect to X of degree at most $\binom{k-1}{2}$, compatible with the linear trinomial factors. That is, P can be written $P=X^{q_1}Q_1+\cdots+X^{q_s}Q_s$ such that $\mu_F(P)=\min_t \mu_F(Q_t)$, and the sum of the degrees of the Q_t's is bounded by $\binom{k-1}{2}$. Inverting the roles of X and Y, one can compute a compatible decomposition of each Q_t with respect to Y. Globally, the polynomial P can be expressed as

$$P=\sum_{t=1}^{s} X^{\alpha(t)} Y^{\beta(t)} R_t$$

where each R_t has ℓ_t terms, $\sum_t \ell_t = k$, and its degree in both X and Y is at most $\binom{\ell_t-1}{2}$. The linear factors of P are the common linear factors of all the R_t's, or equivalently the linear factors of $\gcd(R_1,\ldots,R_s)$. One can thus apply standard algorithms to compute this gcd and then factor it in time polynomial in $\binom{k}{2}$. Moreover, $\mu_F(P)=\mu_F(\gcd(R_1,\ldots,R_s))$. Therefore, this describes an algorithm to compute the linear trinomial factors of P and their multiplicities. $\square$

4.2 Finding multilinear factors

Theorem 4.3. *Let $\mathbb{K}$ be a number field and*

$$P = \sum_{j=1}^{k} a_j X^{\alpha_j} Y^{\beta_j} \in \mathbb{K}[X,Y].$$

There exists a deterministic polynomial time algorithm to compute all the multilinear factors of P, with their multiplicities.

Proof. As for computing linear factors, the task of computing multilinear factors can be split into three parts: computing the obvious monomial factors, computing the binomial factors using Lemma 4.1, and computing factors with at least three monomials.

One can find all factors of the form $uXY + vX + wY + t$ with $wt \neq 0$ using the Gap Theorem for multilinear factors (Theorem 3.4). As for linear factors, one can compute a decomposition of P with respect to both X and Y in the sense of Definition 3.2: $P = X^{\alpha_{(1)}}Y^{\beta_{(1)}}R_1 + \cdots + X^{\alpha_{(s)}}Y^{\beta_{(s)}}R_s$ such that for all $F = uXY + vX + wY + t$ with $wt \neq 0$, $\mu_F(P) = \min(\mu_F(R_1), \dots, \mu_F(R_s))$. Moreover the sum of the degrees of the R_t's (with respect to X or Y) is bounded by $2\binom{k-1}{2}$. Then, finding such factors with their multiplicities is reduced to computing the multilinear factors of $\gcd(R_1, \dots, R_s)$ with multiplicities.

There are two other cases to consider: $t = 0$ and $w = 0$. Both cases can be treated in the same way. Let us first concentrate on the case $t = 0$.

Suppose that $uXY + vX + wY$ divides P, $uvw \neq 0$. Let P_{XY} be the reciprocal polynomial of P with respect to its two variables:

$$P_{XY} = X^A Y^B P(1/X, 1/Y) = \sum_{j=1}^{k} a_j X^{A-\alpha_j} Y^{B-\beta_j},$$

where $A = \max_j \alpha_j$ and $B = \max_j \beta_j$. Then $uXY + vX + wY$ divides P if and only if $u + vY + wX$ divides P_{XY}. Therefore, one can compute the linear factors of P_{XY} with three monomials and deduce the factors of P of the form $uXY + vX + wY$ from them.

One treats in the same way the remaining case $w = 0$. Let $P_X = X^A P(1/X, Y)$, we have that $uXY + vX + t$ divides P if and only if $uY + v + tX$ divides P_X. Therefore, this case is also reduced to the computation of linear factors.

Altogether, this gives a polynomial-time algorithm to compute multilinear factors of bivariate lacunary polynomials. □

In the previous algorithm, factors with at least three monomials are computed in several steps, using different Gap Theorems. There are three cases: the general case $wt \neq 0$, and the special cases $t = 0$ and $w = 0$. Each time, a first decomposition of the polynomial is computed with respect to the variable X, and then refined with respect to the variable Y.

We aim to show that these three distinct steps can be performed in only one step. That is, one can first compute a decomposition with respect to the variable X which is compatible with all the multilinear factors with at least three monomials, and then refine this decomposition with respect to the variable Y.

The following proposition does not necessarily improve the running time of the algorithm and makes it more sequential but it simplifies its implementation. And most importantly, it will be crucial for the generalization to multivariate polynomials.

In the previous algorithms, some order on the monomials has been implicitly used. To compute a decomposition of a polynomial with respect to X, the monomials have to be ordered with an order compatible with the natural order on the exponents of the variable X. In particular, we assume in the next proposition that the indices are ordered such that $j < \ell$ implies $\alpha_j \leq \alpha_\ell$ for all j and ℓ.

Proposition 4.4. *Let P be as in Theorem 4.3. There exists a deterministic polynomial time algorithm to compute a decomposition of P with respect to X, compatible with the set of all multilinear polynomials with at least three monomials, and of degree at most $3\binom{k-1}{2}$.*

Proof. A decomposition $\mathcal{Q} = \{Q_1, \dots, Q_s\}$ is completely determined by a set of indices: for each Q_t, we consider its smallest index j_t. Then $J = \{j_1, \dots, j_s\} \subseteq \{1, \dots, k\}$ determines $\mathcal{Q}$. In the following, the decompositions are represented by their corresponding set of indices. To build a unique decomposition, we take the intersection of several decompositions viewed as sets of indices. If J defines a decomposition compatible with a set $\mathcal{F}$ then by Remark 3.3 any subset $I \subset J$ of indices defines a decomposition of P compatible with $\mathcal{F}$. Therefore, the intersection of two decompositions, respectively compatible with sets $\mathcal{F}$ and $\mathcal{G}$, is compatible with $\mathcal{F} \cup \mathcal{G}$.

It remains to show that the intersection of the decompositions compatible with the different kinds of multilinear factors with at least three monomials has degree at most $3\binom{k-1}{2}$. In the following, we consider only decompositions with respect to X. The decomposition $J = \{j_1, \dots, j_s\}$ compatible with $uXY + vX + wY + t$, $wt \neq 0$, is defined by $j_1 = 1$ and for all $t \geq 2$, j_t is the smallest index such that $\alpha_{j_t} - \alpha_{j_{t-1}} > 2\binom{j_t - j_{t-1}}{2}$.

The decompositions compatible with $uXY + vX + wY$ and $uXY + wY + t$ are the same, that is defined by a same set of indices. Indeed, a decomposition of P compatible with $uXY + vX + wY$ is obtained using the Gap Theorem for linear factors on P_{XY}, while a decomposition of P compatible with $uXY + wY + t$ is obtained using the same Gap Theorem on P_X. In both cases, the first gap is given by the largest ℓ such that $A - \alpha_\ell > A - \alpha_k + \binom{k-\ell}{2}$, that is $\alpha_k - \alpha_\ell > \binom{k-\ell}{2}$. Since both decompositions are obtained using the same equation, they are actually equal.

It remains to prove that the decomposition induced by $J \cap L$ has degree in X at most $3\binom{k-1}{2}$. To this end, we assume that $J \cap L = \emptyset$ and prove that $\alpha_k - \alpha_1 \leq 3\binom{k-1}{2}$. The result then follows by super-additivity of the function $k \mapsto 3\binom{k-1}{2}$. Since $J \cap L = \emptyset$, each gap of L falls strictly between two gaps of J. This implies that the degree of $J \cap L$ is bounded by the sum of the degrees of J and L, that is $3\binom{k-1}{2}$. □

4.3 Generalization to multivariate polynomials

We state the generalization to multivariate polynomials only for multilinear factors. This covers in particular the case of linear factors.

Finding multilinear factors of multivariate polynomials can be performed in three distinct steps as in the case of bivariate polynomials. The first step is the obvious computation of the monomial factors. The second step deals with binomial factors and reduces to univariate factorization. The third step reduces the computation of multilinear factors with at least three monomials to low-degree factorization.

Let us begin with the third step, which is very close to the bivariate case.

Theorem 4.5. *Let $\mathbb{K}$ be an algebraic number field and*

$$P = \sum_{j=1}^{k} a_j X_0^{\alpha_{0,j}} \cdots X_n^{\alpha_{n,j}} \in \mathbb{K}[X_0, \dots, X_n]$$

There exists a deterministic algorithm that finds all the multilinear factors of P with at least three monomials, together with their multiplicities. Its complexity is polynomial in k^n.

A randomized variant of the algorithm achieves a complexity polynomial in k and n. Furthermore, there exists a deterministic algorithm of same complexity for computing the linear factors of P only.

Proof. The idea of the algorithm is as before to compute a decomposition of P, compatible with the set of multilinear factors with at least three monomials. To this end, we first compute a decomposition with respect to the variable X_0, and then refine the decomposition using the other variables, sequentially. The computation of the decomposition for each variable is based on Proposition 4.4.

Without loss of generality, we describe the computation of the decomposition with respect to the variable X_0. Any irreducible multilinear polynomial F with at least three monomials can be written as $F = F_0 X_0 + F_1$ with $F_0, F_1 \in \mathbb{K}[X_1, \dots, X_n]$ and $F_1 \neq 0$. First assume that $F_0 \neq 0$. Since F has at least three monomials, at least one of F_0 and F_1 must have two monomials. If F_0 has at least two monomials, there exists a variable, say X_1, such that $F_0 = uX_1 + v$ for some nonzero $uv \in \mathbb{K}[X_2, \dots, X_n]$. If F_1 has two monomials, $F_1 = wX_1 + t$ for some nonzero $w, t \in \mathbb{K}[X_2, \dots, X_n]$. In both cases, F can be viewed as a polynomial in X_0 and X_1, with at least three monomials, with coefficients in $\mathbb{K}[X_2,\dots,X_n]$. Therefore, one can view P as a bivariate polynomial in X_0 and X_1 over the field $\mathbb{K}(X_2,\dots,X_n)$ and apply Proposition 4.4 to compute a decomposition of P with respect to X_0, compatible with all multilinear polynomials with at least three monomials. In particular, this decomposition is compatible with the multilinear polynomials in X_0, $\dots$, X_n over $\mathbb{K}$ with at least three monomials, and such that the coefficient of X_0 is nonzero. Now the decomposition is also compatible with factors $F \in \mathbb{K}[X_1,\dots,X_n]$ since such factors have to divide every coefficient of P viewed as an element of $\mathbb{K}[X_1,\dots,X_n][X_0]$.

Applying this algorithm sequentially with respect to every variable gives a decomposition of P compatible with all multilinear polynomials with at least three monomials. Its total degree is at most $3\binom{k-1}{2}$ in each variable and the total number of polynomials in the decomposition is bounded by the number k of terms in P. One can therefore compute the gcd of the polynomials in the decomposition and the irreducible factorization of the gcd using classical algorithms. Then we return the multilinear factors, together with their multiplicities. The main computational task of this algorithm is the final irreducible factorization. Using classical algorithms, one can perform this final step in time polynomial in k^n deterministically and in time polynomial in k and n probabilistically [23]. Actually, in the final step one only needs to compute the multilinear factors of the gcd that was computed. Though no deterministic polynomial-time algorithm is known for this computation, a deterministic polynomial-time algorithm is known in the case of linear factors [41]. $\square$

For the computation of binomial multilinear factors, we extend Lemma 4.1. The main difference comes from the fact that in Lemma 4.1 we used the same algorithm four times, once for each possible choice of exponents. In the case of multivariate polynomials, there is an exponential number of choices of exponents. Thus the same strategy yields an algorithm of exponential complexity in the number of variables. Therefore, one has to determine in advance a smaller number of possible vectors of exponents. The proof is inspired by the proof of Kaltofen and Koiran [25, Lemma 5].

Theorem 4.6. *Let $\mathbb{K}$ be an algebraic number field and*

$$P = \sum_{j=1}^{k} a_j X_0^{\alpha_{0,j}} \cdots X_n^{\alpha_{n,j}} \in \mathbb{K}[X_0,\dots,X_n]$$

There exists a deterministic polynomial-time algorithm that finds all the multilinear factors of P with two monomials, together with their multiplicities.

Proof. In this proof, we denote by X the tuple of variables $(X_0,\dots,X_n)$. The strategy is to compute a set of candidate pairs of monomials (X^β, X^γ) such that P may have a factor of the form $uX^\beta + vX^\gamma$, and then to actually compute the factors. For the first step, we write what it means for $uX^\beta + vX^\gamma$ to be a factor of P and deduce conditions on β and γ. The second step is then a reduction to finding linear factors of univariate polynomials.

We begin with the first step. Let F be an irreducible multilinear binomial. One can write

$$F = u\prod_{i=0}^{n} X_i^{\beta_i} + v\prod_{i=0}^{n} X_i^{\gamma_i}$$

with $\beta_i, \gamma_i \in \{0,1\}$ and $\beta_i + \gamma_i \leq 1$ for all i. Without loss of generality, let us assume that $\beta \neq 0$, and let i_0 be an index such that $\beta_{i_0} = 1$. Then F is a factor of P if and only if the polynomial

$$P(X_0, \dots, X_{i_0-1}, -\frac{v}{u}\prod_{i\neq i_0} X_i^{\gamma_i-\beta_i}, X_{i_0+1}, \dots, X_n) \times \prod_{i\neq i_0} X_i^{A_{i_0}}$$

vanishes, where $A_{i_0} = \max_j \alpha_{i_0,j}$. That is, F divides P if and only if

$$\sum_{j=1}^{k} a_j \left(-\frac{v}{u}\right)^{\alpha_{i_0,j}} \prod_{i\neq i_0} X_i^{\alpha_{i,j}+\alpha_{i_0,j}(\gamma_i-\beta_i)+A_{i_0}} = 0.$$

In particular, the term for $j = 1$ has to be canceled out by at least another term. In other words, there must exist $j \in \{2, \dots, k\}$ such that

$$\forall i, \alpha_{i,1} + \alpha_{i_0,1}(\gamma_i - \beta_i) = \alpha_{i,j} + \alpha_{i_0,j}(\gamma_i - \beta_i). \qquad (7)$$

Furthermore $\alpha_{i_0,1} \neq \alpha_{i_0,j}$, for it would imply that $\alpha_{i,1} = \alpha_{i,j}$ for all i otherwise. Thus Equation (7) uniquely determines $\gamma_i - \beta_i$, hence uniquely determines both β_i and γ_i.

The variable X_{i_0} does not play a special role in the previous discussion and the same reasoning applies with any variable X_i such that $\beta_i \neq 0$. In the same way, β and γ play symmetric roles. This means that for every $j \geq 2$, if the system defined by Equation (7) has a nonzero solution (β, γ), it defines a candidate pair (X^β, X^γ) and its symmetric pair (X^γ, X^β). The symmetric pair is redundant and we may only consider the pair (X^β, X^γ). More precisely, for $j \geq 2$, if there exists an integer $q > 0$ such that for all i, either $\alpha_{i,j} - \alpha_{i,1} = 0$ or $\alpha_{i,j} - \alpha_{i,1} = \pm q$, then one can define β_i and γ_i for all i as follows:

$$(\beta_i, \gamma_i) = \begin{cases} (0,0) & \text{if } \alpha_{i,j} - \alpha_{i,1} = 0, \\ (1,0) & \text{if } \alpha_{i,j} - \alpha_{i,1} = q, \\ (0,1) & \text{if } \alpha_{i,j} - \alpha_{i,1} = -q. \end{cases}$$

Therefore, we can compute at most $(k-1)$ candidate pairs. It remains to prove that, given a candidate pair (X^β, X^γ), one can indeed compute all the factors of P of the form $uX^\beta + vX^\gamma$.

The algorithm for the second step is actually almost the same as in the proof of Lemma 4.1. Suppose that $\beta \neq 0$. (Otherwise, invert β and γ.) Let i_0 be any index such that $\beta_{i_0} = 1$. Let

$$Q(Y,X) = \prod_{i \neq i_0} X_i^{A_{i_0}} P(X_0, \dots, X_{i_0-1}, Y \prod_{i \neq i_0} X_i^{\gamma_i - \beta_i}, X_{i_0+1}, \dots, X_n)$$

where $A_{i_0} = \max_j \alpha_{i_0,j}$, viewed as an element of $\mathbb{K}[Y][X_0, \dots, X_n]$. That is, let us write $Q = \sum_\delta q_\delta(Y) X^\delta$.

Let $F = uX^\beta + vX^\gamma$ be a candidate factor. Then F divides P if and only if $G = uY + v$ divides Q, if and only if G divides each q_δ. More precisely, $\mu_F(P) = \mu_G(Q) = \min_\delta \mu_G(q_\delta)$. Therefore, factors of the form $uX^\beta + vX^\gamma$ can be computed in deterministic polynomial-time by computing the factors of the univariate polynomials q_δ, using Lenstra's algorithm [35]. ☐

Note that in the previous algorithm, one could actually decrease the number of candidate pairs. To compute these pairs, we used the fact that the term for $j = 1$ has to be canceled by at least another term. One could then use the same argument for every term: Each term has to be canceled by another one. Applying the same reasoning for all terms would imply more conditions on (β, γ), thus potentially decrease the number of candidate pairs.

4.4 Factorization in other fields of characteristic zero

We have given algorithms to compute the multilinear factors of multivariate polynomials over number fields. Nevertheless, the Gap Theorems these algorithms are based on are valid over any field of characteristic zero. We can thus state our results in a more general form. Our algorithms are actually *reductions* from the multivariate lacunary factorization problem to the univariate lacunary case on the one hand, and the multivariate low-degree case on the other hand. Such kind of reductions is fairly classical in polynomial factorization, see for instance [23].

The proof of the following proposition is the same as the proof of Theorem 4.5.

Proposition 4.7. *Let $\mathbb{K}$ be a field of characteristic 0, and*

$$P = \sum_{j=1}^{k} a_j X_0^{\alpha_{0,j}} \cdots X_n^{\alpha_{n,j}} \in \mathbb{K}[X_0, \dots, X_n].$$

One can compute in deterministic polynomial time a polynomial P_{low} of degree at most $3\binom{k-1}{2}$ such that the sets of multilinear factors (with multiplicities) having at least three monomials of P and P_{low} are equal.

Using this proposition, one can obtain polynomial-time algorithms to compute the multilinear factors with at least three monomials of lacunary polynomials over several fields of characteristic 0. The following corollary gives some examples of fields for which one obtains such algorithms.

Corollary 4.8. *There exist randomized algorithms which given some polynomial $P \in \mathbb{Q}[X_1,\dots,X_n]$, compute its multilinear factors with at least three monomials with coefficients in $\overline{\mathbb{Q}}$, $\mathbb{C}$ and $\mathbb{Q}_p$, respectively. These algorithms run in time polynomial in the size of lacunary representation of P.*

Proof. In all cases, the corollary follows from Proposition 4.7 using a factorization algorithm that is polynomial in the degree of the input polynomial. For references to the numerous algorithms for the factorization over $\overline{\mathbb{Q}}$ or $\mathbb{C}$, we refer the reader to [10]. An algorithm for the factorization over $\mathbb{Q}_p$ is due to Chistov [9]. □

To compute binomial factors, we need a univariate lacunary factorization algorithm. To the best of our knowledge, algorithms for this task are only known over algebraic number fields. Note that over algebraic closed fields, there is no hope of computing these factors in polynomial time: The number of univariate factors equals the degree of the lacunary polynomial and is thus exponential in the size of the input.

4.5 Comparison with previous work

We aim to compare our techniques with the ones of Kaltofen and Koiran [24, 25]. Their first result deals with linear factors of bivariate lacunary polynomials over the rational numbers, while the second ones is an extension to the case of low-degree factors, with multiplicities, of multivariate lacunary polynomials over number fields. Thus, our algorithm applies to more fields: nonzero characteristic (see Section 5) or absolute factorization for instance. On the other hand, it is limited to multilinear factors rather than arbitrary low-degree factors.

Let us now compare the techniques used and the complexities of both algorithms. For the sake of simplicity, we focus on the following case: computing linear factors of bivariate polynomials over the integers or the rationals. We first briefly highlight the main differences and then perform a somewhat detailed complexity analysis of both algorithms.

Binomial factors are computed essentially in the same way in both algorithms, using Lenstra's algorithm for univariate lacunary polynomials [35]. Let us then focus on the task of finding trinomial factors. The high-level structure of both algorithms is the same. Both of them identify gaps in the exponent vector of the input polynomial, and use these gaps to write the polynomial as a sum of low-degree polynomials. The linear factors of the input polynomial are then the common linear factors of all these low-degree polynomials. Two differences make our algorithm simpler: First our computation of the gaps is based on purely combinatorial considerations on the exponents while one needs to compute the height of the polynomial in Kaltofen and Koiran's algorithm; Secondly our algorithm directly gives the multiplicities of the factors while their algorithm has to be applied to the successive (sparse) derivatives of the input polynomial in order to get the multiplicities.

To analyse the complexities of both algorithms, we remark that the main computational task is in both cases the final step when the common linear factors of the low-degree polynomials must be computed. For this step, one can either compute the gcd of the low-degree polynomials and its linear factors, or compute the linear factors of one of the low-degree polynomials and then use a divisibility test on the other polynomials. Whichever strategy one chooses, a relevant measure to analyse the complexity of any of the two algorithms is the size of the low-degree polynomials. Their coefficients are coefficients of the input polynomial. Their degree is related to the Gap Theorems. In our case, Theorem 3.1 directly shows that the low-degree polynomials have degree at most $O(k^2)$ where k is the number of terms of the input polynomial. On the other hand, Kaltofen and Koiran [24] prove a gap of size $O(\log(k)+h)$ between consecutive powers, where h is the (logarithmic) height of the input polynomial, which translates into a bound $O(k(\log k+h))$ on the degree of the low-degree polynomials to be factored. In the case of integer coefficients, the height is bounded by the bitsize b of the coefficients. To compute the height in the case of rational coefficients, one first needs to reduce all coefficients to the same denominator and the height is bounded by $O(kb)$. In other words, the bound on the degree of the low-degree polynomials to be factored in Kaltofen and Koiran's algorithm is $O(kb)$ (ignoring logarithmic factors) for integer coefficients and $O(k^2b)$ for rational coefficients.

The best general factorization algorithm for degree-d integer bivariate polynomials with coefficients of bitsize b has complexity $O(d^6b^5)$ [34, 22]. To compute the sole linear factors, the best strategy has the much lower complexity $O(d^2b)$ (see [6] for a complete complexity analysis). For our algorithm, this means that the complexity of the factorization step is bounded by $O(k^4b)$ in the case of integer coefficients. In Kaltofen and Koiran's algorithm, the same step runs in time $O(k^2b^3)$ (ignoring logarithmic factors in k). Since computing the multiplicity of the factors in their approach requires to apply the same algorithm to at most $(k-1)$ derivatives of the polynomial, the total complexity is $O(k^3b^3)$. In other word, our algorithm is faster as soon as $b \geq \sqrt{k}$ in the case of integer coefficients. To compute the linear factors of rational polynomials, the coefficients have to be reduced to the same denominator and the coefficient size can increase by a factor d. That is, computing the linear factors of rational polynomials has complexity $O(d^3b)$. Using the bound of the previous paragraph and the same reasoning as above, the complexity of our algorithm becomes $O(k^6b)$ and the complexity of Kaltofen and Koiran's algorithm becomes

$O(k^7 b^4)$. Thus the worst-case complexity of our algorithm is better for rational coefficients.

5 Positive characteristic

To extend the previous results to positive characteristic, one needs an equivalent of Theorem 2.1. Unfortunately, Theorem 2.1 does not hold in positive characteristic. In characteristic 2, the polynomial $(1+X)^{2^n} + (1+X)^{2^{n+1}} = X^{2^n}(X+1)$ has only two terms, but its valuation equals 2^n. Therefore, its valuation cannot be bounded by a function of the number of terms. Note that this can be generalized to any positive characteristic. In characteristic p, one can consider the polynomial $\sum_{i=1}^{p}(1+X)^{p^{n+i}}$.

Nevertheless, the exponents used in all these examples depend on the characteristic. In particular, the characteristic is always smaller than the largest exponent that appears. We shall show that in large characteristic, Theorem 2.1 still holds and can be used to give factorization algorithms. This contrasts with the previous results that use the notion of height of an algebraic number, hence are not valid in any positive characteristic.

In fact, Theorem 2.1 holds as soon as $\operatorname{wr}(f_1, \dots, f_k)$ does not vanish. The difficulty in positive characteristic is that it is not true anymore that the Wronskian does not vanish as soon as $(f_j)_j$ is a linearly independent family. Consider for instance the family $f_1 = 1$ and $f_2 = X^2$ in characteristic 2. Yet, the Wronskian is still related to linear independence by the following result, see [28]:

Proposition 5.1. *Let $\mathbb{K}$ be a field of characteristic p and $f_1, \dots, f_k \in \mathbb{K}[X]$. Then f_1, $\dots$, f_k are linearly independent* over $\mathbb{K}[X^p]$ *if and only if their Wronskian does not vanish.*

This allows us to give an equivalent of Theorem 2.1 in large positive characteristic.

Theorem 5.2. *Let $P = \sum_{j=1}^{k} a_j X^{\alpha_j}(1+X)^{\beta_j} \in \mathbb{K}[X]$ with $\alpha_1 \le \cdots \le \alpha_k$. If the characteristic p of $\mathbb{K}$ satisfies $p > \max_j(\alpha_j + \beta_j)$, then the valuation of P is at most $\max_j(\alpha_j + \binom{k+1-j}{2})$, provided P is nonzero.*

Proof. Let $f_j = X^{\alpha_j}(1+X)^{\beta_j}$ for $1 \le j \le k$. The proof of Theorem 2.1 has two steps: We prove that we can assume that the Wronskian of the f_j's does not vanish, and under this assumption we get a bound of the valuation of the polynomial. The second part only uses the non-vanishing of the Wronskian and can be used here too. We are left with proving that the Wronskian of the f_j's can be assumed to be nonzero when the characteristic is large enough.

Assume that the Wronskian of the f_j's is zero: By Proposition 5.1, there is a vanishing linear combination of the f_j's with coefficients b_j in $\mathbb{K}[X^p]$. Let us write

$b_j = \sum b_{i,j} X^{ip}$. Then $\sum_i X^{ip} \sum_j b_{i,j} f_j = 0$. Since $\deg f_j = \alpha_j + \beta_j < p$, $\sum_j b_{i,j} f_j = 0$ for all i. We have thus proved that there is a linear combination of the f_j's equal to zero with coefficients in $\mathbb{K}$. Therefore, we can assume we have a basis of the f_j's whose Wronskian is nonzero and use the same argument as for the characteristic zero. □

Based on this result, the algorithms we develop in characteristic zero for PIT and factorization can be used for large enough characteristics. Computing with lacunary polynomials in positive characteristic has been shown to be hard in many cases [43, 29, 31, 24, 2, 26]. In particular, Bi et al. [2] have recently shown that it is NP-hard to find roots in $\mathbb{F}_p$ for polynomials over $\mathbb{F}_p$.

Let $\mathbb{F}_{p^s}$ be the field with p^s elements for p a prime number and $s > 0$. It is represented as $\mathbb{F}_p[\xi]/\langle\varphi\rangle$ where φ is a monic irreducible polynomial of degree s with coefficients in $\mathbb{F}_p$. As for number fields, φ can be given as input of the algorithms, and a coefficient $c \in \mathbb{F}_{p^s}$ is represented by a polynomial of degree smaller than $\deg(\varphi)$.

Theorem 5.3. *Let $\mathbb{F}_{p^s}$ be a finite field, and*

$$P = \sum_{j=1}^{k} a_j X^{\alpha_j} (uX + v)^{\beta_j} \in \mathbb{F}_{p^s}[X],$$

where $p > \max_j(\alpha_j + \beta_j)$. There exists a deterministic algorithm to test if P vanishes identically, whose complexity is polynomial in the bitsizes of the α_j's and β_j's, in k and $s \log p$.

Proof idea. The proof of this theorem is very similar to the proof of Theorem 3.5, using Theorem 5.2 instead of Theorem 2.1. The main difference occurs when $u = 0$ or $v = 0$. In these cases, we rely in characteristic zero on Lenstra's algorithm to test sums of the form $\sum_j a_j v^{\beta_j}$ for zero. There is no equivalent of Lenstra's algorithm in positive characteristic, but these tests are actually much simpler. These sums can be evaluated using repeated squaring in time polynomial in $\log(\beta_j)$, that is polynomial in the input length.

The basic operations in the algorithm are operations in the ground field $\mathbb{F}_p$. Therefore, the result also holds if bit operations are considered. The only place where computations in $\mathbb{F}_{p^s}$ have to be performed in the algorithm is the tests for zero of coefficients of the form $\sum_j \binom{\alpha_j}{\ell_j} a_j u^{-\alpha_j} (-v)^{\ell_j}$ where the α_j's and ℓ_j's are integers and $a_j \in \mathbb{F}_{p^s}$, and the sum has at most k terms. The binomial coefficient is to be computed *modulo* p using for instance Lucas' Theorem [36]. □

Note that the condition $p > \max_j(\alpha_j + \beta_j)$ means that p has to be greater than the degree of P. This condition is a fairly natural condition for many algorithms dealing with polynomials over finite fields, especially prime fields, for instance for root finding algorithms [2].

We now turn to the problem of factoring lacunary polynomials with coefficients in fields of large characteristic. We state it in the most general case of finding multilinear factors of multivariate polynomials.

Theorem 5.4. *Let $\mathbb{F}_{p^s}$ be the field with p^s elements, and*

$$P = \sum_{j=1}^{k} a_j X_0^{\alpha_{0,j}} \cdots X_n^{\alpha_{n,j}} \in \mathbb{F}_{p^s}[X_0, \ldots, X_n],$$

where $p > \max_j(\alpha_j + \beta_j)$. There exists a probabilistic polynomial-time algorithm to find all the multilinear factors of P with at least three monomials, together with their multiplicities.

On the other hand, deciding whether P has a binomial factor is NP*-hard under randomized reductions. More precisely, for every pair of relatively prime multilinear monomials (X^β, X^γ), deciding whether there exist nonzero u and v such that $uX^\beta + vX^\gamma$ divides P is* NP*-hard under randomized reductions.*

Proof. The second part of the theorem is the consequence of the NP-hardness (under randomized reductions) of finding roots in $\mathbb{F}_{p^s}$ of lacunary univariate polynomials with coefficients in $\mathbb{F}_{p^s}$ [31, 2, 26]: Let Q be a lacunary univariate polynomial over $\mathbb{F}_{p^s}$, and let $d = \deg(Q)$. Let us define $P(X_0, \ldots, X_n) = (X^\beta)^d Q(X^{\gamma-\beta})$ where $X^{\gamma-\beta} = \prod_i X_i^{\gamma_i - \beta_i}$. Then P is a polynomial. We aim to show that Q has a nonzero root if and only if P has a binomial factor of the form $uX^\beta + vX^\gamma$. Let $F = uX^\beta + vX^\gamma$. Without loss of generality, we can assume that $\beta \neq 0$ and $\beta_0 = 1$. Then F divides P if and only if

$$(X^\gamma)^{\max_j \alpha_{0,j}} P\left(-\frac{v}{u}\prod_{i>0} X_i^{\gamma_i-\beta_i}, X_1, \ldots, X_n\right) = 0.$$

Let $X^\delta = (X^\gamma)^{\max_j \alpha_{0,j}} (X^\beta)^d$. Since $\beta_0 = 1$ and $\gamma_0 = 0$, the previous equality is equivalent to

$$X^\delta Q\left(\left(-\frac{v}{u}\prod_{i>0} X_i^{\gamma_i-\beta_i}\right)^{-1} \prod_{i>0} X_i^{\gamma_i-\beta_i}\right) = X^\delta Q(-\frac{u}{v}) = 0.$$

In other words, this is equivalent with the fact that $-u/v$ is a root of Q. Deciding whether $uX^\beta + vX^\gamma$ divides P is thus NP-hard under randomized reductions.

For the first part, the algorithm we propose is actually the same as in characteristic zero (Theorem 4.2). This means that it relies on known results for factorization of dense polynomials. Yet, the only polynomial-time algorithms known for factorization in positive characteristic are probabilistic [42]. Therefore our algorithm is probabilistic and not deterministic as in characteristic zero. ☐

References

[1] F. Amoroso, L. Leroux, and M. Sombra. Overdetermined systems of sparse polynomial equations. *Found. Comput. Math.*, 15(1):53–87, 2015. DOI:10.1007/s10208-014-9207-y. Presented at MEGA 2013.

[2] J. Bi, Q. Cheng, and J. M. Rojas. Sub-Linear Root Detection, and New Hardness Results, for Sparse Polynomials Over Finite Fields. In *Proc. ISSAC*, pages 61–68. ACM, 2013. DOI:10.1145/2465506.2465514. ARXIV:1204.1113.

[3] F. Bihan, J. M. Rojas, and C. E. Stella. Faster real feasibility via circuit discriminants. In *Proc. ISSAC*, pages 39–46. ACM, 2009. DOI:10.1145/1576702.1576711.

[4] M. Bläser, M. Hardt, R. J. Lipton, and N. K. Vishnoi. Deterministically testing sparse polynomial identities of unbounded degree. *Inform. Process. Lett.*, 109 (3):187–192, Jan. 2009. DOI:10.1016/j.ipl.2008.09.029.

[5] A. Bostan and P. Dumas. Wronskians and linear independence. *Am. Math. Mon.*, 117(8):722–727, 2010. DOI:10.4169/000298910X515785.

[6] A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, and B. Salvy. *Algorithmes Efficaces en Calcul Formel*. 2017. URL https://hal.archives-ouvertes.fr/AECF/.

[7] A. Chattopadhyay, B. Grenet, P. Koiran, N. Portier, and Y. Strozecki. Factoring bivariate lacunary polynomials without heights. In *Proc. ISSAC*, pages 141–158. ACM, 2013. DOI:10.1145/2465506.2465932.

[8] Q. Cheng. Derandomization of sparse cyclotomic integer zero testing. In *Proc. FOCS*, pages 74–80. IEEE, 2007. DOI:10.1109/FOCS.2007.15.

[9] A. Chistov. Algorithm of polynomial complexity for factoring polynomials over local fields. *J. Math. Sci.*, 70(4):1912–1933, 1994. DOI:10.1007/BF02112431.

[10] G. Chèze and A. Galligo. Four lectures on polynomial absolute factorization. In A. Dickenstein and I. Z. Emiris, editors, *Solving Polynomial Equations*, volume 14 of *Algorithms Comput. Math.*, pages 339–392. 2005. DOI:10.1007/3-540-27357-3_9.

[11] F. Cucker, P. Koiran, and S. Smale. A polynomial time algorithm for Diophantine equations in one variable. *J. Symb. Comput.*, 27(1):21–30, 1999. DOI:10.1006/jsco.1998.0242.

[12] M. Filaseta, A. Granville, and A. Schinzel. Irreducibility and Greatest Common Divisor Algorithms for Sparse Polynomials. In J. McKee and C. Smyth, editors, *Number Theory and Polynomials*, volume 352 of *Lond. Math. S.*, pages 155–176. Camb. U. Press, 2008.

[13] M. A. Forbes, R. Saptharishi, and A. Shpilka. Hitting Sets for Multilinear Read-once Algebraic Branching Programs, in Any Order. In *Proc. STOC'14*, pages 867–875. ACM, 2014. DOI:10.1145/2591796.2591816.

[14] M. Giesbrecht and D. S. Roche. On lacunary polynomial perfect powers. In *Proc. ISSAC*, pages 103–110. ACM, 2008. DOI:10.1145/1390768.1390785.

[15] M. Giesbrecht and D. S. Roche. Detecting lacunary perfect powers and computing their roots. *J. Symb. Comput.*, 46(11):1242 – 1259, 2011. DOI:10.1016/j.jsc.2011.08.006.

[16] B. Grenet. Computing low-degree factors of lacunary polynomials: a Newton-Puiseux approach. In *Proc. ISSAC*, pages 224–231. ACM, 2014. DOI:10.1145/2608628.2608649.

[17] B. Grenet. Lacunaryx: Computing bounded-degree factors of lacunary polynomials. *ACM Comm. Comput. Algebra*, 49(4):121–124, 2015. DOI:10.1145/2893803.2893807. ISSAC 2015 Software Presentation.

[18] B. Grenet. Bounded-degree factors of lacunary multivariate polynomials. *J. Symb. Comput.*, 75:171–192, 2016. DOI:10.1016/j.jsc.2015.11.013. ISSAC 2014 special issue.

[19] B. Grenet, P. Koiran, N. Portier, and Y. Strozecki. The Limited Power of Powering: Polynomial Identity Testing and a Depth-four Lower Bound for the Permanent. In *Proc. FSTTCS*, number 13 in LIPIcs, pages 127–139. LZI, 2011. DOI:10.4230/LIPIcs.FSTTCS.2011.127.

[20] D. Grigoriev and M. Karpinski. A zero-test and an interpolation algorithm for the shifted sparse polynomials. In *Proc. AAECC-10*, pages 162–169. Springer, 1993. DOI:10.1007/3-540-56686-4_41.

[21] G. Hajós. [solution to problem 41] (in hungarian). *Mat. Lapok*, 4:40–41, 1953.

[22] W. Hart, M. van Hoeij, and A. Novocin. Practical polynomial factoring in polynomial time. In *Proc. ISSAC*, pages 163–170. ACM, 2011. DOI:10.1145/1993886.1993914.

[23] E. Kaltofen. Polynomial-Time Reductions from Multivariate to Bi- and Univariate Integral Polynomial Factorization. *SIAM J. Comput.*, 14(2):469–489, 1985. DOI:10.1137/0214035.

[24] E. Kaltofen and P. Koiran. On the complexity of factoring bivariate supersparse (lacunary) polynomials. In *Proc. ISSAC*, pages 208–215. ACM, 2005. DOI:10.1145/1073884.1073914.

[25] E. Kaltofen and P. Koiran. Finding small degree factors of multivariate supersparse (lacunary) polynomials over algebraic number fields. In *Proc. ISSAC*, pages 162–168. ACM, 2006. DOI:10.1145/1145768.1145798.

[26] E. L. Kaltofen and G. Lecerf. Factorization of Multivariate Polynomials. In G. L. Mullen and D. Panario, editors, *Handbook of Finite Fields*, Disc. Math. Appl., chapter 11 (sec. 5). CRC Press, 2013. ISBN 9781439873786.

[27] E. L. Kaltofen and M. Nehring. Supersparse black box rational function interpolation. In *Proc. ISSAC*, pages 177–186. ACM, 2011. DOI:10.1145/1993886.1993916.

[28] I. Kaplansky. *An introduction to differential algebra*. Actualités scientifiques et industrielles. Hermann, 1976. ISBN 9782705612511.

[29] M. Karpinski and I. Shparlinski. On the computational hardness of testing square-freeness of sparse polynomials. In *Proc. AAECC-13*, pages 731–731. Springer, 1999. DOI:10.1007/3-540-46796-3_47.

[30] N. Kayal and C. Saha. On the Sum of Square Roots of Polynomials and Related Problems. In *Proc. CCC*, pages 292–299. IEEE, 2011. DOI:10.1109/CCC.2011.19.

[31] A. Kipnis and A. Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *Proc. CRYPTO*, pages 19–30. Springer, 1999. DOI:10.1007/3-540-48405-1_2.

[32] P. Koiran, N. Portier, and S. Tavenas. A Wronskian approach to the real τ-conjecture. *J. Symb. Comput.*, 68:195–214, 2015. DOI:10.1016/j.jsc.2014.09.036. Presented at MEGA 2013.

[33] Y. N. Lakshman and B. D. Saunders. Sparse shifts for univariate polynomials. *Appl. Alg. Eng. Comm. Comput.*, 7(5):351–364, Sept. 1996. DOI:10.1007/BF01293594.

[34] G. Lecerf. New recombination algorithms for bivariate polynomial factorization based on Hensel lifting. *Appl. Alg. Eng. Comm. Comput.*, 21(2):151–176, 2010. DOI:10.1007/s00200-010-0121-5.

[35] H. W. Lenstra. Finding small degree factors of lacunary polynomials. In K. Győry, H. Iwaniec, and J. Urbanowicz, editors, *Number theory in progress*. 1999. URL http://hdl.handle.net/1887/2154.

[36] É. Lucas. Théorie des fonctions numériques simplement périodiques. *Amer. J. Math.*, 1(2–4):184–196,197–240,289–321, 1878. DOI:10.2307/2369308 (part 1), DOI:10.2307/2369311 (part 2), DOI:10.2307/2369373 (part 3).

[37] H. Montgomery and A. Schinzel. Some arithmetic properties of polynomials in several variables. In A. Baker and D. W. Masser, editors, *Transcendence Theory: Advances and Applications*, chapter 13, pages 195–203. Academic Press, 1977.

[38] M. Petkovšek, H. S. Wilf, and D. Zeilberger. *A=B*. AK Peters, 1996. ISBN 978-1568810638.

[39] D. Plaisted. Sparse complex polynomials and polynomial reducibility. *J. Comput. Syst. Sci.*, 14(2):210–221, 1977. DOI:10.1016/S0022-0000(77)80013-5.

[40] A. Shpilka and A. Yehudayoff. *Arithmetic Circuits: A Survey of Recent Results and Open Questions*. Foundations and trends in theoretical computer science. Now, 2010. ISBN 978-1-60198-400-5.

[41] I. Volkovich. Deterministically Factoring Sparse Polynomials into Multilinear Factors and Sums of Univariate Polynomials. In *Proc. APPROX-RANDOM*, volume 40 of *LIPIcs*. LZI, 2015. DOI:10.4230/LIPIcs.APPROX-RANDOM.2015.943.

[42] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Camb. U. Press, 3rd edition, 2013. ISBN 9781107039032.

[43] J. von zur Gathen, M. Karpinski, and I. Shparlinski. Counting curves and their projections. *Comput. Complex.*, 6(1):64–99, 1996. DOI:10.1007/BF01202042.