Finding small degree factors of lacunary polynomials

H.W. Lenstra, Jr.

To Andrzej Schinzel

Abstract. If K is an algebraic number field of degree at most m over the field $\mathbf{Q}$ of rational numbers, and $f \in K[X]$ is a polynomial with at most k non-zero terms and with $f(0) \neq 0$, then for any positive integer d the number of irreducible factors of f in $K[X]$ of degree at most d, counted with multiplicities, is bounded by a constant that depends only on m, k, and d. This is proved in a companion paper (H.W. Lenstra, Jr., "On the factorization of lacunary polynomials"). In the present paper an algorithm for actually finding those factors is presented. The algorithm assumes that K is specified by means of an irreducible polynomial h with integral coefficients and leading coefficient 1, such that $K = \mathbf{Q}(\alpha)$ for a zero α of h. Also, the polynomial $f = \sum_i a_i X^i$ is supposed to be given in its sparse representation, i.e., as the list of pairs (i, a_i) for which $a_i \neq 0$, each a_i being represented by means of its vector of coefficients on the vector space basis 1, α, $\ldots$, $\alpha^{(\deg h)-1}$ of K over $\mathbf{Q}$. If l denotes the "length" of these input data, when written out in binary, then the running time of the algorithm, measured in bit operations, is at most $(l+d)^c$ for some absolute and effectively computable constant c. Taking $K = \mathbf{Q}$ and $d = 1$, one deduces that all rational zeroes of a sparsely represented polynomial with rational coefficients can be found in polynomial time. This answers a question raised by F. Cucker, P. Koiran, and S. Smale.

1991 Mathematics Subject Classification: Primary 11R09, 11Y16.

Key words: lacunary polynomial, computational complexity.

Acknowledgements. The author was supported by NSF under grant No. DMS 92-24205. He thanks J.A. Csirik, C.J. Smyth, and J.D. Vaaler for helpful assistance.

1. Introduction

F. Cucker, P. Koiran, and S. Smale [2] exhibited a polynomial time algorithm accomplishing the following. Suppose that a polynomial $f = \sum_i a_i X^i$ in one variable with coefficients in the ring $\mathbf{Z}$ of integers is specified in its *sparse* representation, i.e., by the list of pairs (i, a_i) for which $a_i \neq 0$. Then the algorithm finds all zeroes of f in $\mathbf{Z}$. One of the questions they raised is whether one can also find all *rational* zeroes of f in polynomial time. In the present paper I show that this is indeed the case. Rational zeroes correspond to irreducible factors of degree 1 over the field $\mathbf{Q}$ of rational numbers, and my result extends to finding irreducible factors of low degrees over algebraic number fields.

For a ring R, let $R[X]$ denote the ring of polynomials in one variable X over R. A polynomial is *monic* if its leading coefficient is 1.

Theorem. *There is a deterministic algorithm that, for some positive real number c, has the following property: given an algebraic number field K, a sparsely represented non-zero polynomial $f \in K[X]$, and a positive integer d, the algorithm finds all monic irreducible factors of f in $K[X]$ of degree at most d, as well as their multiplicities, and it spends time at most $(l+d)^c$, where l denotes the length of the input data.*

The conventions in this theorem are as in [8, Section 2]. Rational numbers are represented as fractions of integers. An algebraic number field K is supposed to be specified by means of a monic irreducible polynomial $h \in \mathbf{Z}[Y]$ such that $K = \mathbf{Q}(\alpha)$ for a zero α of h; an element of K, such as a coefficient of f, is then represented by means of its vector of coefficients on the vector space basis $(\alpha^j)_{j=0}^{m-1}$ of K over $\mathbf{Q}$, where $m = \deg h$. Here the polynomial $h = \sum_{j=0}^{m} h_j Y^j$ is *densely* represented, i.e., by means of the list of all pairs (j, h_j), $0 \le j \le m$, including those for which $h_j = 0$. The *length* (or the *size*) of the input data is defined in [8, 2.1] (cf. [2, Sec. 1]); it may informally be thought of as the number of bits needed to spell out the data in binary. The *time* taken by an algorithm is measured in bit operations.

One way of finding the irreducible factors of f in $K[X]$ is first to convert f from its sparse to its dense representation and next to apply one of the well-known polynomial time algorithms (see [4; 6]) for factoring densely represented polynomials over algebraic number fields. This procedure, however, fails to satisfy the time bound stated in the theorem. Consider, for example, the case in which $f = X^n - 1$ for large n, with fixed d and K; then the length l of the data has order of magnitude $\log n$, and the length of the dense representation of f, which is about n, is exponential in l, so it cannot be written down within time $(l+d)^c$.

Our result is "uniform in K": rather than having a separate algorithm for each K, we have one single algorithm that accepts data specifying K as part of the input; for fixed d, the running time is polynomially bounded in terms of the length of these data and the data specifying f. For varying d, the running time can still be said to be polynomially bounded in terms of the length of the input data and the possible length of the output, since the polynomials produced by the algorithm are densely represented and may have degree up to d. However, the algorithm may spend time exponential in $\log d$ and still find no factors.

The number of different factors found by the algorithm is at most an absolute constant times $k^2 \cdot 2^n \cdot n \cdot \log(2nk)$, where k is the number of non-zero terms of f and $n = d \cdot [K : \mathbf{Q}]$, by [9, Theorem 1]. This is an exponential bound, but it is completely independent of the degree and the coefficients of f and of the coefficients of the polynomial defining K.

The idea behind the algorithm is best illustrated on an easier problem. Suppose that a sparsely represented polynomial $f \in \mathbf{Q}[X]$ as well as a rational number x are given. How does one test in polynomial time whether or not $f(x)$ vanishes? Just substituting x for X in f is not feasible, since if the degree of f is very large then $f(x)$ may be too large to write down, let alone to calculate. Fortunately, if it is just a matter of testing whether $f(x)$ vanishes, one can get away with a much simpler procedure. Namely, excluding the easy cases $x = \pm 1$, one proves that a large degree polynomial with not many non-zero terms can vanish in x only if it does so for obvious reasons, namely if there are "widely" spaced non-negative integers u and "low" degree polynomials f_u with $f_u(x) = 0$ and $f = \sum_u f_u \cdot X^u$. The bounds that make this statement valid depend on the number of non-zero terms of f and on the sizes of the numerators and denominators of its coefficients, but they do not depend on x. Thus, to test whether f vanishes at a given rational number $x \neq \pm 1$, one "breaks" f into appropriate polynomials f_u and one tests whether they all vanish at x.

The algorithm underlying our theorem follows the same idea, and it is presented in Section 4. The basic result justifying the procedure (Proposition 2.3) is formulated and proved in Section 2. Section 3 contains several auxiliary algorithms, one of which finds the cyclotomic factors of f. The phenomenon that these require separate treatment is familiar from Schinzel's work on factors of lacunary polynomials.

Should the need for finding small degree factors of sparse polynomials over algebraic number fields ever arise, then a suitable variant of my method may very well have practical value; however, as it stands it is designed only to lead to a valid and efficient proof of the theorem.

Several results in this paper assert the existence of algorithms with certain properties. In each case, such an algorithm is actually exhibited in the paper itself or in one of the references. All these algorithms are deterministic, and the constants appearing in running time estimates are effectively computable. Polynomials are densely represented in algorithms, unless it is explicitly stated that they are sparsely represented.

By $\mathbf{R}$ we denote the field of real numbers, and by $\mathbf{C}$ the field of complex numbers. The degree of a field extension $E \subset F$ is written $[F : E]$. The multiplicative group of non-zero elements of a field F is denoted by F^*.

2. Heights and lacunary polynomials

Let $\bar{\mathbf{Q}}$ denote an algebraic closure of $\mathbf{Q}$, and let $K \subset \bar{\mathbf{Q}}$ be a finite extension of $\mathbf{Q}$. Write M_K for the set of non-trivial prime divisors of K, and for each $v \in M_K$, let $\|\cdot\|_v : K \to \mathbf{R}$ be a corresponding valuation; we assume that these valuations are normalized as in [5, Chap. 2, Sec. 2]. This normalization is characterized by the facts that the *product formula*

$$\prod_{v \in M_K} \|x\|_v = 1 \qquad \text{for all } x \in K^* \tag{2.1}$$

holds, and that the *relative height function*

$$H_K : K \to \mathbf{R}, \qquad H_K(x) = \prod_{v \in M_K} \max\{1, \|x\|_v\}$$

(see [5, Chap. 3, Sec. 1]) satisfies $H_K(k) = k^{[K:\mathbf{Q}]}$ for all positive integers k.

The *absolute height function* $H : \bar{\mathbf{Q}} \to \mathbf{R}$ is defined by

$$H(x) = H_K(x)^{1/[K:\mathbf{Q}]},$$

where K is such that $x \in K$; this is independent of the choice of K. For example, for r, $s \in \mathbf{Z}$, $s > 0$, $\gcd(r,s) = 1$ one has $H(r/s) = \max\{|r|, s\}$.

For a positive integer n, we define

$$c(n) = \frac{2}{n \cdot \bigl(\log(3n)\bigr)^3} \quad \text{if } n \geq 2,$$

and $c(1) = \log 2$. This is a decreasing function of n.

Proposition 2.2. *Let n be a positive integer. Suppose that $x \in \bar{\mathbf{Q}}^*$ is of degree at most n over $\mathbf{Q}$, and that $\log H(x) < c(n)$. Then x is a root of unity.*

Proof. See [12, Corollary 2]. This proves 2.2.

If K is as above, then for $v \in M_K$ we extend $\|\cdot\|_v$ to a function $K[X] \to \mathbf{R}$ by $\left\|\sum_i a_i X^i\right\|_v = \max_i \|a_i\|_v$. Define $\mathbf{H} : \bar{\mathbf{Q}}[X] \to \mathbf{R}$ by $\mathbf{H}(f) = \prod_{v \in M_K} \|f\|_v^{1/[K:\mathbf{Q}]}$, where K is chosen such that $f \in K[X]$; this is independent of the choice.

Proposition 2.3. *Let k, t, u be non-negative integers, and let $f \in \bar{\mathbf{Q}}[X]$ be a polynomial with at most $k+1$ non-zero terms. Suppose that n is a positive integer with*

$$u - t > \frac{\log\bigl(k \cdot \mathbf{H}(f)\bigr)}{c(n)},$$

and that f is written as the sum of two polynomials g, $h \in L[X]$ such that every non-zero term of g has degree at most t and every non-zero term of h has degree at least u. Then every zero of f in $\bar{\mathbf{Q}}^*$ that has degree at most n over $\mathbf{Q}$ and that is not a root of unity is a common zero of g and h.

Proof. Let $x \in \bar{\mathbf{Q}}^*$ be of degree at most n over $\mathbf{Q}$, and suppose that $f(x) = 0$. Then we have $g(x) = -h(x)$. We shall assume that $g(x) = -h(x) \neq 0$, and prove that x is a root of unity.

Let K be chosen such that $x \in K$ and $f \in K[X]$. Then we have g, $h \in K[X]$. Let $v \in M_K$. From $h(x) \neq 0$ it follows that h has at least 1 non-zero term, and since f has at most $k+1$ non-zero terms it follows that g has at most k non-zero terms. Thus $g(x)$ is a sum of at most k terms $a_i x^i$, with $\|a_i\|_v \leq \|f\|_v$ and $i \leq t$. This leads to the estimate

$$\|g(x)\|_v \leq \max\{1, \|k\|_v\} \cdot \|f\|_v \cdot \|x\|_v^t \qquad \text{if } \|x\|_v \geq 1.$$

Likewise, $h(x)$ is a sum of at most k terms $a_i x^i$, with $\|a_i\|_v \leq \|f\|_v$ and $i \geq u$, so

$$\|h(x)\|_v \leq \max\{1, \|k\|_v\} \cdot \|f\|_v \cdot \|x\|_v^u \qquad \text{if } \|x\|_v \leq 1.$$

We have $\|g(x)\|_v = \|h(x)\|_v$, so we can combine these two statements in

$$\max\{1, \|x\|_v\}^{u-t} \cdot \|g(x)\|_v \leq \max\{1, \|k\|_v\} \cdot \|f\|_v \cdot \|x\|_v^u.$$

Raise this to the power $1/[K : \mathbf{Q}]$ and take the product over $v \in M_K$. Using the fact that $H(k) = k$, and applying (2.1) to x and to $g(x)$ (which are both supposed to be non-zero), one finds that

$$H(x)^{u-t} \leq k \cdot \mathbf{H}(f).$$

By hypothesis, we have $k \cdot \mathbf{H}(f) < \exp\bigl((u-t)c(n)\bigr)$. It follows that $\log H(x) < c(n)$, so 2.2 implies that x is a root of unity. This proves 2.3.

Proposition 2.4. *Let $K \subset \bar{\mathbf{Q}}$ be a finite extension of $\mathbf{Q}$, and let $f \in K[X]$. Let r be a positive integer such that all coefficients of rf are algebraic integers, and let s be a positive real number with the property that for every field homomorphism $\sigma : K \to \mathbf{C}$ and every coefficient a of f one has $|\sigma a| \leq s$. Then one has $\mathbf{H}(f) \leq rs$.*

Proof. First assume that $r = 1$. Then each coefficient of f is an algebraic integer, so $\|f\|_v \leq 1$ for each non-archimedean $v \in M_K$. Also, by definition of s we have $\|f\|_v \leq s$ for each real $v \in M_K$, and $\|f\|_v \leq s^2$ for each complex $v \in M_K$. Collecting all v, one obtains $\mathbf{H}(f) \leq s$, since the number of real v plus twice the number of complex v equals $[K : \mathbf{Q}]$. The case $r > 1$ is reduced to the case $r = 1$ by the formula $\mathbf{H}(rf) = \mathbf{H}(f)$, which follows from (2.1), applied to $x = r$. This proves 2.4.

3. Auxiliary algorithms

Proposition 3.1. *There is an algorithm that, for some positive constant c_1, has the following property: given an algebraic number field K and a densely represented non-zero polynomial $f \in K[X]$, the algorithm finds the complete factorization of f into monic irreducible factors in $K[X]$, and it does so in time at most l^{c_1}, where l denotes the length of the data.*

For the proof of this proposition, and a description of the algorithm, we refer to [4; 6]. It makes use of lattice basis reduction [7].

Let K be a field of characteristic zero. For $f \in K[X]$, we define the *sparse derivative* $f^{[1]}$ of f to be the ordinary derivative of f/X^t, if X^t is the highest power of X dividing f, and we define it to be 0 if $f = 0$; the higher sparse derivatives $f^{[i]}$ are defined inductively by $f^{[i]} = (f^{[i-1]})^{[1]}$, and for convenience we set $f^{[0]} = f$. If $f \neq 0$, then clearly the number of non-zero terms of $f^{[1]}$ is one less than the number of non-zero terms of f. It follows that $f^{[i]} = 0$ if and only if i is greater than or equal to the number of non-zero terms of f.

Proposition 3.2. *Let K be a field of characteristic zero, let $f \in K[X]$ be a non-zero polynomial, and let $g \in K[X]$ be an irreducible polynomial with $g(0) \neq 0$. Then the number of factors of g in f is equal to $\min\{i \geq 0 : g$ does not divide $f^{[i]}\}$, and it is smaller than the number of non-zero terms of f.*

Proof. The first assertion is proved in a routine manner by induction on the number of factors of g in f. If f has exactly $k+1$ terms, then $f^{[k]}$ is a polynomial with exactly one term, which is not divisible by g. Thus the second assertion follows from the first. This proves 3.2.

The second assertion can also be derived from an observation of Hajós (see [3; 11, Lemma 1]).

Proposition 3.3. *There is an algorithm that, for some positive constant c_2, has the following property. Given an algebraic number field K and a sparsely represented non-zero polynomial $f \in K[X]$, the algorithm computes the sparse representations of the sparse derivatives $f^{[i]}$ for all $i \geq 0$ that are less than the number of non-zero terms of f; and it does so in time at most l^{c_2}, where l denotes the length of the data.*

Proof. This is obvious—one just computes the polynomials $f^{[i]}$ directly from the definition. This proves 3.3.

Proposition 3.4. *There is an algorithm that, for some positive constant c_3, has the following property: given an algebraic number field K, a sparsely represented non-zero polynomial $f \in K[X]$, and a positive integer r, the algorithm computes the greatest common divisor of f and $X^r - 1$ in $K[X]$, and it does so in time at most $(l+r)^{c_3}$, where l denotes the length of the data.*

Proof. The algorithm runs as follows. Let $f = \sum_i a_i X^{t(i)}$. For each i, compute the remainder $u(i)$ of $t(i)$ upon division by r. Next compute the polynomial $h = \sum_i a_i X^{u(i)}$, and use the Euclidean algorithm for polynomials in order to compute the greatest common divisor of h with $X^r - 1$. This gcd is the output of the algorithm.

To prove the correctness, it suffices to remark that from $t(i) \equiv u(i) \bmod r$, for each i, it follows that $f \equiv h \bmod X^r - 1$, and therefore $\gcd(f, X^r - 1) = \gcd(h, X^r - 1)$.

The running time estimate is proved in a completely straightforward way; note that h is densely represented, and has degree less than r. For a running time estimate of the Euclidean algorithm for polynomials, see [4, Cor. 1.8]. This proves Proposition 3.4.

If K is a field, we call a polynomial $g \in K[X]$ *cyclotomic* if, for some positive integer r, it is a monic irreducible factor of $X^r - 1$ in $K[X]$.

Proposition 3.5. *There is an algorithm that, for some positive constant c_4, has the following property: given an algebraic number field K, a sparsely represented* *non-zero polynomial $f \in K[X]$, and a positive integer d, the algorithm computes in time at most $(l+d)^{c_4}$ all cyclotomic factors g of f in $K[X]$ that have degree at most d, as well as, for each such g, the multiplicity $m(g)$ of g as a factor of f; here l denotes the length of the input data.*

Proof. We claim that the following algorithm has the stated properties. It produces a list of pairs g, $m(g)$, which is initially supposed to be empty.

For each integer $r = 1, 2, \ldots, 2 \cdot (d \cdot [K : \mathbf{Q}])^2$ in succession, do the following. Compute $\gcd(f, X^r - 1)$ with the algorithm of 3.4, factor $\gcd(f, X^r - 1)$ into irreducible factors in $K[X]$ by means of the algorithm of 3.1, and discard those irreducible factors that appear already on the list or have degree greater than d. Adjoin the remaining irreducible factors g to the list, and for each of them compute $m(g)$ from the formula

$$m(g) = \min\{i : 1 \leq i \leq k,\ g \text{ does not divide } \gcd(f^{[i]}, X^r - 1)\},$$

where k is one less than the number of non-zero terms of f; here $f^{[i]}$ is computed in its sparse representation by the algorithm of 3.3, and its gcd with $X^r - 1$ is computed in its dense representation as in 3.4.

This completes the description of the algorithm.

The proof of the bound for the running time is straightforward, and left to the reader. We prove that each cyclotomic factor g of f of degree at most d is found by the algorithm, and that $m(g)$ is its multiplicity. Let g be such a factor, let ζ be a zero of g in an extension field of K, and let r be the multiplicative order of ζ. Denoting the Euler φ-function by φ, we have

$$\begin{aligned}\varphi(r) = [\mathbf{Q}(\zeta) : \mathbf{Q}] \leq [K(\zeta) : \mathbf{Q}] = [K(\zeta) : K] \cdot [K : \mathbf{Q}]\\ = (\deg g) \cdot [K : \mathbf{Q}] \leq d \cdot [K : \mathbf{Q}].\end{aligned}$$

The elementary inequality $\varphi(r) \geq \sqrt{r/2}$ now implies that $r \leq 2 \cdot (d \cdot [K : \mathbf{Q}])^2$. Therefore g is indeed found by the algorithm. From Proposition 3.2 it follows that $m(g)$ equals the multiplicity of g as a factor of f. This proves 3.5.

The function $\mathbf{H}$ in the following result is as defined in Section 2, with $\bar{\mathbf{Q}}$ equal to an algebraic closure of $\mathbf{Q}$ that contains K.

Proposition 3.6. *There is an algorithm that, for some positive constant c_5, has the following property: given an algebraic number field K and a sparsely represented non-zero polynomial $f \in K[X]$, the algorithm computes in time at most l^{c_5} a positive integer b satisfying $b \geq k \cdot \mathbf{H}(f)$; here k is 1 less than the number of non-zero terms of f, and l denotes the length of the input data.*

Proof. As in the introduction, it is assumed that K is specified by means of an irreducible polynomial $h = \sum_{j=0}^{m} h_j Y^j \in \mathbf{Z}[Y]$, with $h_m = 1$, with the property that $K = \mathbf{Q}(\alpha)$ for some zero α of h. Also, each coefficient a_i of f is supposed to be represented by a vector $(q_{ij})_{j=0}^{m-1}$ with $q_{ij} \in \mathbf{Q}$ for which $a_i = \sum_{j=0}^{m-1} q_{ij}\alpha^j$. For each field homomorphism $\sigma : K \to \mathbf{C}$, the complex number $\sigma\alpha$ is a zero of h and therefore satisfies $|\sigma\alpha| \leq B = \sum_{j=0}^{m-1} |h_j|$. Hence if r is a positive integer for which

$r \cdot q_{ij} \in \mathbf{Z}$ for all i and j, then one has

$$|\sigma(r \cdot a_i)| \leq s_i = \sum_{j=0}^{m-1} |r \cdot q_{ij}| \cdot B^j$$

for all field homomorphisms $\sigma : K \to \mathbf{C}$ and all i. Thus, by 2.4 the number $b = k \cdot \max_i s_i$ is a positive integer satisfying $b \geq k \cdot \mathbf{H}(f)$. One can compute b in polynomial time in a straightforward way, taking for r the least common multiple (or even the product) of the denominators of the q_{ij}. This proves 3.6.

4. Proof of the theorem

The proof of the theorem stated in the introduction consists of three parts: the description of the algorithm underlying the theorem, the proof of its correctness, and the running time estimate.

To describe the algorithm, let an algebraic number field K, a sparsely represented non-zero polynomial $f \in K[X]$, and a positive integer d be given. The algorithm produces a list of pairs g, $m(g)$, which is initially supposed to be empty.

Step 1. *Find the cyclotomic factors.* Use the algorithm of 3.5 to find all cyclotomic factors g of f in $K[X]$, as well as their multiplicities $m(g)$.

Step 2. *Compute a bound for the gap width.* Let $k+1$ be the number of non-zero terms of f. Use the algorithm of 3.3 to compute $f^{[i]}$ for $0 \leq i < k$ in their sparse representations. Next, applying the algorithm of 3.6 to each $f^{[i]}$, compute positive integers b_i satisfying

$$b_i \geq (k-i) \cdot \mathbf{H}(f^{[i]}) \qquad \text{for } i = 0, 1, \ldots, k-1.$$

Finally, compute a positive integer b satisfying

$$b \geq \frac{\max\{\log b_i : 0 \leq i < k\}}{c(d \cdot [K : \mathbf{Q}])} > b - 2,$$

with the function c as defined in Section 2. For the logarithms, one can use the algorithms in [1]. (For the significance of $b-2$, see [10, Sec. 1, end].)

Step 3. *Split f at the big gaps.* Let $f = \sum_{t \in T} a_t X^t$, where T is a set of $k+1$ non-negative integers and $a_t \in K^*$ for each $t \in T$. Ordering T, determine the subset $U = \{u \in T : \text{there does not exist } t \in T \text{ with } u - b \leq t < u\}$ of T, where b is as computed in Step 2. Next, for each $u \in U$, determine the subset $T(u) = \{t \in T : u = \max\{v \in U : v \leq t\}\}$ of T. (Then T is the disjoint union of the sets $T(u)$, for $u \in U$, and each $T(u)$ contains u.) To conclude this step, compute the polynomials

$$f_u = \sum_{t \in T(u)} a_t X^{t-u} \qquad (u \in U),$$

in their dense representations. (These polynomials satisfy $f_u(0) \neq 0$ and $f = \sum_{u \in U} f_u \cdot X^u$.)

Step 4. *Factor a dense polynomial.* Using the Euclidean algorithm for polynomials (see [4, Cor. 1.8]), compute $h = \gcd_{u\in U} f_u$. Factor h into monic irreducible factors in $K[X]$ by means of the algorithm of 3.1.

Step 5. *Assemble the results.* Discard each monic irreducible factor of h that occurs already among the factors computed in Step 1 or has degree greater than d. Adjoin each of the remaining monic irreducible factors g of h to the list, with $m(g)$ equal to the multiplicity of g as a factor of h. Finally, if 0 does not belong to the set T of Step 3, adjoin $g = X$ to the list, with $m(X)$ equal to the smallest element of T.

This concludes the description of the algorithm.

We next prove the correctness. The parenthetical statements in Step 3 are readily verified. The polynomial h divides each f_u, so it divides f. One deduces that the polynomials g produced by the algorithm are indeed monic irreducible factors of f in $K[X]$ of degree at most d. Also, h is not divisible by X, since none of the f_u is, so from Step 5 one sees that no g is produced twice.

Conversely, let g be a monic irreducible factor of f in $K[X]$ of degree at most d. We prove that g is produced by the algorithm, and that $m(g)$ equals the multiplicity of g as a factor of f. These statements are obvious if g is cyclotomic (Step 1) and if $g = X$ (Step 5). In the other case, let $\bar{\mathbf{Q}}$ be an algebraic closure of $\mathbf{Q}$ containing K, and let $x \in \bar{\mathbf{Q}}$ be a zero of g. By hypothesis, x is not a root of unity, and $x \neq 0$. The degree $[\mathbf{Q}(x) : \mathbf{Q}]$ of x over $\mathbf{Q}$ satisfies

$$[\mathbf{Q}(x) : \mathbf{Q}] \leq [K(x) : \mathbf{Q}] = [K(x) : K] \cdot [K : \mathbf{Q}] = (\deg g) \cdot [K : \mathbf{Q}] \leq d \cdot [K : \mathbf{Q}].$$

For each $u \in U$, we now apply 2.3 with $n = d \cdot [K : \mathbf{Q}]$, and with

$$\sum_{v\in U,\, v<u} f_v \cdot X^v, \qquad \sum_{v\in U,\, v\geq u} f_v \cdot X^v$$

in the roles of g and h. From

$$\frac{\log\bigl(k \cdot \mathbf{H}(f)\bigr)}{c(n)} \leq \frac{\log b_0}{c(d \cdot [K : \mathbf{Q}])} \leq b$$

and the definitions of U and f_u it follows that the inequality of 2.3 is satisfied. Now 2.3 asserts that x is a zero of both polynomials just displayed. Since this is the case for each $u \in U$, one infers that $f_u(x) = 0$ for all $u \in U$, and therefore that $h(x) = 0$. Hence g is an irreducible factor of h, and it is produced by the algorithm. To show that $m(g)$ is the multiplicity of g in f, we repeat the argument just given with $f^{[i]}$ in the role of f, for each $i = 1, 2, \ldots, k-1$. The representation $f = \sum_{u\in U} f_u X^u$ induces a similar representation of each $f^{[i]}$. Thanks to the choice of b we can still apply (2.3). Using 3.2, one deduces that x is a j-fold zero of f if and only if it is a j-fold zero of each f_u, the case $j > k$ being vacuously correct. Thus, the multiplicity of g as a factor of f is the same as the multiplicity $m(g)$ of g as a factor of $h = \gcd_u f_u$. This proves the correctness of the algorithm.

We prove the running time estimate. Since each b_i, in Step 2, is computed by a polynomial time algorithm, its logarithm is bounded by a constant power of the length l of the data. Also, from the definition of $c(n)$ in Section 2 one sees that

$1/c(n)$ is bounded by a constant times n^2. It follows that the bound b computed in Step 2 is bounded by a constant power of $l+d$. Now let $u \in U$. The definitions of U and $T(u)$ imply that any two consecutive non-zero terms of f_u have degrees differing by at most b. Since f_u has at most $k+1$ non-zero terms, one of which has degree 0, it follows that $\deg f_u \leq k \cdot b$. Therefore the length of the dense representation of f_u is bounded by a constant power of $l+d$. This implies that the time taken by the polynomial time operations on the f_u in Step 4 remains within the bound stated in the theorem. It is a routine matter to prove that this also applies to the time taken by the other steps of the algorithm.

This proves the theorem stated in the introduction.

References

[1] Brent, R.P., Fast multiple-precision evaluation of elementary functions. J. Assoc. Comput. Mach. 23 (1976), 242–251.

[2] Cucker, F., Koiran, P., Smale, S., A polynomial time algorithm for diophantine equations in one variable. J. Symbolic Comput., to appear.

[3] Hajós, G., [Solution to problem 41] (in Hungarian). Mat. Lapok 4 (1953), 40–41.

[4] Landau, S., Factoring polynomials over algebraic number fields. SIAM J. Comput. 14 (1985), 184–195.

[5] Lang, S., Fundamentals of diophantine geometry. Springer, New York 1983.

[6] Lenstra, A.K., Factoring polynomials over algebraic number fields. In: Computer algebra (ed. by J.A. van Hulzen; Lecture Notes in Comput. Sci. 162), 245–254. Springer, Berlin 1983.

[7] Lenstra, A.K., Lenstra, H.W., Jr., Lovász, L., Factoring polynomials with rational coefficients. Math. Ann. 261 (1982), 515–534.

[8] Lenstra, H.W., Jr., Algorithms in algebraic number theory. Bull. Amer. Math. Soc. (N.S.) 26 (1992), 211–244.

[9] — On the factorization of lacunary polynomials. This volume, 277–291.

[10] Lenstra, H.W., Jr., Pomerance, C., A rigorous time bound for factoring integers. J. Amer. Math. Soc. 5 (1992), 483–516.

[11] Montgomery, H.L., Schinzel, A., Some arithmetic properties of polynomials in several variables. In: Transcendence theory: advances and applications (ed. by A. Baker, D.W. Masser), Chapter 13, 195–203. Academic Press, London 1977.

[12] Voutier, P., An effective lower bound for the height of algebraic numbers. Acta Arith. 74 (1996), 81–95.