Polynomial-time algorithms for the factorization of polynomials

Polynomial-time algorithms for the factorization of polynomials

Academisch Proefschrift

ter verkrijging van de graad van
doctor in de Wiskunde en Natuurwetenschappen
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
dr. D.W. Bresters
hoogleraar in de Faculteit
der Wiskunde en Natuurwetenschappen
in het openbaar te verdedigen
in de Aula der Universiteit
(tijdelijk in de Lutherse Kerk, ingang Singel 411, hoek Spui)
op woensdag 16 mei 1984 des namiddags te 4.00 uur

door

Arjen Klaas Lenstra

geboren te Groningen

1984
Centrum voor Wiskunde en Informatica, Amsterdam

Promotor: Dr. P. van Emde Boas

Preface

This thesis consists of an introductory part and seven papers. In these papers polynomial-time algorithms are presented for the factorization of various types of polynomials. The algorithms are based on L. Lovász' *basis reduction algorithm*, described in the first section of the second paper, combined with a technique that was introduced in the first paper.

I owe many thanks to my promotor P. van Emde Boas and to the coauthors of the second paper, H.W. Lenstra, Jr. and L. Lovász. The Centrum voor Wiskunde en Informatica and its publishing department are gratefully acknowledged, in particular R.T. Baanders and D. Zwarst. Valuable suggestions were made by J.K. Lenstra and G.M. Tuynman. Finally, I want to thank A.S. van Dobbenburgh for her support.

Amsterdam, January 1984 Arjen K. Lenstra

Contents

1. Introduction

In 1982 a polynomial-time algorithm for factoring polynomials in one variable with rational coefficients was published [II]. This *L^3-algorithm* came as a rather big surprise: hardly anybody expected that the problem allowed solution in polynomial time. The purpose of this introductory part is to present an informal description of the L^3-algorithm.

To measure the complexity of our algorithms we have to specify the encoding of the polynomials to be factored. Two *encoding schemes* for polynomials can be distinguished, a *dense* encoding scheme and a *sparse* encoding scheme. If a polynomial is densely encoded, all its coefficients, including the zeros, are listed; in a sparse encoding only the non-zero coefficients are listed. Here we use the dense encoding scheme. This implies that an algorithm to factor polynomials runs in polynomial time if for any polynomial f to be factored, the running time is bounded by a fixed polynomial function of the degrees and the size of the coefficients of f.

After the introduction of some basic tools in Section 2, we describe in Section 3 a well-known older algorithm to factor polynomials, the Berlekamp-Hensel algorithm, and we will indicate why this algorithm is not polynomial-time. Roughly speaking, the reason is that the irreducible factors we are looking for (which will frequently be called the *true factors*) are determined by a combinatorial search among other, p-adic factors.

A true factor can also be regarded as a short vector in a certain integral lattice, a concept that was introduced in [I]. Therefore we consider the problem of computing short vectors in a lattice in Section 4, and thereafter we explain the L^3-algorithm in Section 5.

This same technique of looking for short vectors can be applied to other polynomial factoring problems as well. Some of these generalizations of the

L^3-algorithm are presented in Section 6. We conclude in Section 7 with some remarks about the relative merits of these polynomial-time algorithms for the factorization of polynomials.

2. Preliminaries

In the subsequent sections the following three notions will play an important role: Berlekamp's algorithm, Hensel's lemma, and Mignotte's bound. These are the basic tools for most of the polynomial factoring algorithms. We will briefly explain here what they stand for.

Berlekamp's algorithm is an algorithm to determine the irreducible factors of a polynomial in one variable with coefficients in a finite field. Let $\mathbb{F}_q$ denote a finite field containing q elements, for some prime power $q = p^m$, and let f be a polynomial in $\mathbb{F}_q[X]$ of degree n. To factor f, the maximal number of additions, multiplications, and divisions in $\mathbb{F}_q$ to be carried out by Berlekamp's algorithm is $O(p\,m\,n^3)$. This is the best worst-case running time that is known for an algorithm to factor polynomials in $\mathbb{F}_q[X]$. There exist *probabilistic* algorithms for which the *expected* running time is linear in $\log p$ rather than linear in p, as is the case in Berlekamp's algorithm. Although such methods are usually much faster in practice, no upper bound can be given for their worst-case running time, and therefore they are irrelevant for our purposes. For a description of Berlekamp's algorithm we refer to [1; 11].

The Hensel lemma can be formulated as follows. Let p be a prime number, and let k be a positive integer. By $\mathbb{Z}/p^k\mathbb{Z}$ we will denote the ring of integers modulo p^k. Suppose that a polynomial $f \in \mathbb{Z}[X]$ and a factor $h \in (\mathbb{Z}/p^k\mathbb{Z})[X]$ of $f \bmod p^k$ in $(\mathbb{Z}/p^k\mathbb{Z})[X]$ are given, such that $h \bmod p$ and $(f \bmod p)/(h \bmod p)$ are relatively prime in $(\mathbb{Z}/p\mathbb{Z})[X]$ and such that h has leading coefficient equal to one (notice that, because p is prime, $\mathbb{Z}/p\mathbb{Z}$ is a finite field containing p elements). Hensel's lemma guarantees the existence of a unique polynomial $a \in (\mathbb{Z}/p\mathbb{Z})[X]$ of degree smaller than degree(h), such that $h + p^k a \in (\mathbb{Z}/p^{k+1}\mathbb{Z})[X]$ is a factor of $f \bmod p^{k+1}$ in $(\mathbb{Z}/p^{k+1}\mathbb{Z})[X]$. Furthermore, its proof gives an algorithm to construct this a, and this algorithm needs a number of bit operations that is bounded by a polynomial function of degree(f) and $\log p^k$ (cf. [11: exercise 4.6.2.22; 22; 23]). Thus, Hensel's lemma enables us to extend or *lift* a factorization of $f \bmod p$ to a factorization of $f \bmod p^k$ for any $k \in \mathbb{Z}_{>0}$ that we want. This computation can be done in poly-

nomial time as long as k is polynomially bounded. Such a factorization of $f \bmod p^k$ will be called a *p-adic factorization of precision* p^k.

Finally, Mignotte's bound is an upper bound for the coefficients of a factor of a polynomial in one variable with integral coefficients. Let $f = \sum_{i=0}^{n} f_i X^i \in \mathbb{Z}[X]$ be a polynomial of degree n, and let $g = \sum_{i=0}^{m} g_i X^i \in \mathbb{Z}[X]$ be a factor of degree m of f in $\mathbb{Z}[X]$. Mignotte has proved in [14] that

$$(2.1) \qquad |g_i| \leq \binom{m}{i} |f|,$$

where $|f|$ denotes the *length* $(\sum_{i=0}^{n} f_i^2)^{\frac{1}{2}}$ of the polynomial f. It follows that

$$(2.2) \qquad |g| \leq \binom{2m}{m}^{\frac{1}{2}} |f|.$$

Notice that $\log|g| = O(n + \log|f|)$, so that the length of a dense encoding of a factor is polynomially bounded by the length of a dense encoding of the polynomial itself. Similar bounds for polynomials in more than one variable can be found in [6].

In the next section we will see that Berlekamp's algorithm, Hensel's lemma, and Mignotte's bound together give rise to an important algorithm to factor polynomials in $\mathbb{Z}[X]$, the Berlekamp-Hensel algorithm.

3. The Berlekamp-Hensel algorithm

The Berlekamp-Hensel algorithm was the first practical algorithm to factor polynomials in one variable with integral coefficients. In this section we present one of the simplest versions of this algorithm and we discuss its most important properties. Although many improvements of the algorithm have been suggested by several authors, the basic ideas remained the same, and hence we will ignore these variants. Also we will not discuss the generalizations of the Berlekamp-Hensel algorithm to polynomials in more than one variable.

The Berlekamp-Hensel algorithm essentially works as follows. First, a sufficiently precise p-adic factorization of the polynomial to be factored is

computed. Next, the true factors are determined by combining these p-adic factors in the proper way. We now present a somewhat more detailed description of the algorithm.

Let $f \in \mathbb{Z}[X]$ of degree n be the polynomial to be factored. For simplicity we assume that f is *monic*, i.e. f has leading coefficient one. The first step of the Berlekamp-Hensel algorithm is to remove the multiple factors from f. Because a factor of multiplicity $k \geq 1$ in f has multiplicity $k-1$ in the derivative f' of f, this can be done by dividing f by $\gcd(f,f')$, where this gcd can be computed by means of one of the subresultant algorithms (cf. [2]). So, from now on we may assume that f is *square-free*, i.e. f does not contain multiple factors.

Next, we determine a prime number p such that the polynomial $f \bmod p$ in $(\mathbb{Z}/p\mathbb{Z})[X]$ is square-free in $(\mathbb{Z}/p\mathbb{Z})[X]$. This condition on p is equivalent to the condition that p does not divide the *discriminant* $\mathrm{discr}(f) \in \mathbb{Z}_{\neq 0}$ of f (notice that $\mathrm{discr}(f) \neq 0$ because f is square-free). This implies that such a prime number p indeed exists, and that p can be bounded by a polynomial function of n and $\log|f|$.

In the third step we apply Berlekamp's algorithm to compute the complete factorization of $f \bmod p$ in $(\mathbb{Z}/p\mathbb{Z})[X]$. We may assume that the factors of $f \bmod p$ are monic. Clearly the irreducible factors of f in $\mathbb{Z}[X]$ are also factors of $f \bmod p$, but these factors of $f \bmod p$ are not necessarily irreducible in $(\mathbb{Z}/p\mathbb{Z})[X]$. So, the set of irreducible factors of $f \bmod p$ can be partitioned into a number of subsets such that each subset corresponds to an irreducible factor of f in $\mathbb{Z}[X]$. That is, the product of the elements of such a subset is just an irreducible factor of f in $\mathbb{Z}[X]$, reduced modulo p. Thus, these subsets will in general not be sufficient to reconstruct the factors of f in $\mathbb{Z}[X]$, because the coefficients of the resulting product are only integers modulo p. Therefore, before we look for the proper combinations of the p-adic factors, we first have to compute the p-adic factorization of f up to a higher precision.

This is what we do in the next step, where we apply Hensel's lemma, which is possible because $f \bmod p$ is square-free and because we assumed that the factors of $f \bmod p$ are monic. We modify each irreducible factor $\tilde{h}$ of $f \bmod p$ in $(\mathbb{Z}/p\mathbb{Z})[X]$ into a factor h of $f \bmod p^k$ in $(\mathbb{Z}/p^k\mathbb{Z})[X]$, for a value of $k \in \mathbb{Z}_{>0}$ that we will specify below. The polynomials h are monic, so that the factorization in $(\mathbb{Z}/p^k\mathbb{Z})[X]$ that we find in this way is unique.

The value of k has to be chosen in such a way that the coefficients of the combinations that we will have to consider are not too small. Therefore, if we represent $\mathbb{Z}/p^k\mathbb{Z}$ by $\{-[(p^k-1)/2], \ldots, -1, 0, 1, \ldots, [(p^k)/2]\}$, then k has to be chosen such that $(p^k-1)/2$ is greater than the largest possible coefficient of any factor of f in $\mathbb{Z}[X]$. From (2.1) it follows that we can take k minimal such that $(p^k-1)/2 > \binom{n}{n/2}|f|$.

Now that we have a sufficiently precise p-adic factorization of f, we are ready for the last step of the Berlekamp-Hensel algorithm, the determination of the true factors of f. As explained above, this can be done by looking at combinations of p-adic factors. So, for all subsets of our set of p-adic factors, we test whether the product of the p-adic factors in a subset is a true factor of f in $\mathbb{Z}[X]$, until all irreducible factors of f in $\mathbb{Z}[X]$ are found. Of course, we have to arrange the subsets in order of increasing cardinality, to guarantee that the factors of f that we find are irreducible.

This completes the description of the Berlekamp-Hensel algorithm. Let us consider its running time. It is not difficult to verify that, up to the last step, everything can be done in polynomial time. Unfortunately, this is not the case for the last step, the search for the true factors.

Namely, suppose that f is irreducible in $\mathbb{Z}[X]$ and that the number of irreducible factors of $f \bmod p$ is r. Then we have to look at all subsets of cardinality at most $r/2$, before we are sure that none of them yields a factor of f in $\mathbb{Z}[X]$. This implies that the number of subsets to be considered is exponential in r, and because the degree n of f is the only a priori upper bound that we can give for r, we get a bound on the running time of the last step, that is exponential in n.

In [7] a method is given to generate infinite classes of irreducible polynomials in $\mathbb{Z}[X]$, such that, for some $c \in \mathbb{R}_{>0}$ and for every prime number p, the number of p-adic factors is at least cn. For these polynomials the number of subsets to be considered is indeed exponential in n, so that the exponential-time bound that we derived is the best possible.

In practice however, the situation is not so bad as it seems, and the algorithm usually has no problems to factor high-degree polynomials with large coefficients. Also, in [4] it is made plausible that, under certain assumptions concerning the distribution of the degrees of the factors of f, the expected number of subsets to be considered in the last step is at most n^2. This is in accordance with the practical experience that the last step

usually takes much less time than the computation of the p-adic factorization.

Clearly, to obtain a polynomial-time algorithm to factor polynomials in $\mathbb{Z}[X]$, we have to invent another method to reconstruct the factors in $\mathbb{Z}[X]$ from the p-adic factors. In Section 5 we will see that only one sufficiently precise p-adic factor suffices to reconstruct the corresponding irreducible factor in $\mathbb{Z}[X]$, so that we do not have to look for the proper combination of p-adic factors anymore. Before we can explain this construction and show that it gives a polynomial-time factoring algorithm, we have to present an important result concerning integral lattices; this will be done in the next section.

4. Short vectors in lattices

Let $b_1, b_2, \ldots, b_n \in \mathbb{Z}^n$ be linearly independent. The n-dimensional *lattice* $L \subset \mathbb{Z}^n$ with basis $b_1, b_2, \ldots, b_n$ is defined as the set of integral linear combinations of the vectors $b_1, b_2, \ldots, b_n$:

$$L = \{\Sigma_{i=1}^{n} r_i b_i : r_i \in \mathbb{Z}\}.$$

In Section 5 we will be interested in determining short vectors in a lattice, where we use the ordinary Euclidean norm for vectors. We will not give any detailed algorithms here; we will only briefly sketch what is known about computing short vectors in a lattice, and mention an important recent result.

Until now, no polynomial-time algorithm is known to compute a shortest non-zero vector in a lattice (polynomial-time means here polynomial-time in n and the size of the entries of the vectors b_i). In fact, the problem is widely conjectured to be NP-hard, but this has not yet been proved. (If we replace the L_2-norm by the L_∞-norm, then the shortest vector problem is known to be NP-hard (cf. [16]).) At several places more or less practical algorithms to calculate shortest vectors are presented [5; 11; 15]. Although the running times of these algorithms are not analyzed, they are certainly not polynomial-time. Also, in general they perform quite poorly for high-dimensional lattices (say $n \geq 15$). If we fix the dimension n of the lattice however, then a shortest vector can be found in polynomial time. This is a consequence of the

polynomial-time algorithm to solve integer linear programming problems with a fixed number of variables [13].

In 1981, L. Lovász invented an important algorithm, the so-called *basis reduction algorithm*, which made it possible to compute reasonably short vectors in a lattice in polynomial time. More precisely, this algorithm computes a non-zero vector b, belonging to a basis for L, such that $|b| \leq 2^{(n-1)/2}|x|$, for every $x \in L$, $x \neq 0$, where $||$ denotes the Euclidean length. So, in polynomial time we can find a vector in the lattice that is no more than $2^{(n-1)/2}$ times longer than a shortest vector in the lattice. In fact the basis reduction algorithm does not only compute an approximation of a shortest vector. It also computes a so-called *reduced basis* for the lattice, which is, roughly speaking, a basis that is reasonably orthogonal (cf. Section 7). A detailed description of the algorithm and a careful analysis of its running time are given in [II].

There are special cases in which the basis reduction algorithm computes a shortest vector. This will happen for instance if all vectors that are linearly independent of the shortest vector, are more than $2^{(n-1)/2}$ times longer than the shortest vector. This situation will occur in Section 5.

Unfortunately, the fact that the basis reduction algorithm runs in polynomial time does not imply that it is very fast in practice, although it is much better than the algorithms mentioned above. If $B \in \mathbb{Z}_{>0}$ bounds the number of bits in the coordinates of the basis $b_1, b_2, \ldots, b_n$ for L, then a theoretical bound on the number of bit operations to be performed is $O(n^6 B^3)$ (cf. [II]; a slightly better bound, namely $O(n^6 B^2 + n^5 B^3)$, was derived in [9]). Experiments by A.M. Odlyzko showed that in practice the running time is proportional to nB^3. To give an impression of actual running times, we conclude this section with some results from Odlyzko's implementation on a Cray-1 computer:

n	B	average running time in minutes
31	55	0.5
31	65	0.75
41	70	1.2
51	88	3
51	180	18
71	140	14
81	160	23

5. The L^3-algorithm

We return to the problem of factoring polynomials in $\mathbb{Z}[X]$. We will show that the basis reduction algorithm enables us to formulate a polynomial-time algorithm to factor polynomials in $\mathbb{Z}[X]$. Again, let $f \in \mathbb{Z}[X]$ of degree n be the polynomial to be factored. We will assume that the gcd of the coefficients of f is 1 (i.e. f is *primitive*), and that f is square-free.

From Section 2 we know that it is possible to compute a p-adic factorization of f up to any precision that we want, where the prime number p is chosen in such a way that $f \bmod p$ is square-free in $(\mathbb{Z}/p\mathbb{Z})[X]$. Let, for some positive integer k, the polynomial $h \in (\mathbb{Z}/p^k\mathbb{Z})[X]$ be a monic p-adic factor of $f \bmod p^k$, such that $h \bmod p$ is irreducible in $(\mathbb{Z}/p\mathbb{Z})[X]$. It follows that there exists a unique irreducible factor h_0 of f in $\mathbb{Z}[X]$, such that h divides $h_0 \bmod p^k$ in $(\mathbb{Z}/p^k\mathbb{Z})[X]$. We will see that, if k is chosen sufficiently large, then the p-adic factor h suffices to determine the factor h_0 of f (of course, if f is irreducible, we will find $h_0 = f$).

The L^3-algorithm proceeds roughly as follows. First, we construct a lattice such that the factor h_0 that we are looking for is contained in this lattice, and such that h_0 is much shorter than the other vectors in the lattice. Next, we determine h_0 by means of the basis reduction algorithm, which is possible because h_0 is very short.

We will explain how to construct a basis for such a lattice. Denote by ℓ the degree of h; so $0 < \ell \leq n$. Let us suppose that the degree of h_0 is m, with $\ell \leq m \leq n$. To start with, we know that h divides $h_0 \bmod p^k$ in $(\mathbb{Z}/p^k\mathbb{Z})[X]$. This implies that h_0 belongs to the set L of polynomials of degree at most m which have h as a factor, when taken modulo p^k. Because h was chosen to be monic, we have that

$$(5.1) \qquad L = \{p^k r + h\,q \,:\, r,\, q \in \mathbb{Z}[X],\ \text{degree}(r) < \ell,\ \text{degree}(q) \leq m-\ell\}.$$

We define the polynomials $b_0, b_1, \ldots, b_m \in \mathbb{Z}[X]$ as $b_i = p^k X^i$ for $0 \leq i < \ell$ and $b_i = h\,X^{i-\ell}$ for $\ell \leq i \leq m$. With (5.1) this gives

$$L = \mathbb{Z}\,b_0 + \mathbb{Z}\,b_1 + \ldots + \mathbb{Z}\,b_{\ell-1} + \mathbb{Z}\,b_\ell + \mathbb{Z}\,b_{\ell+1} + \ldots + \mathbb{Z}\,b_m,$$

so that we can rewrite (5.1) as $L = \{\Sigma_{i=0}^{m} r_i\,b_i \,:\, r_i \in \mathbb{Z}\}$.

Regarding the polynomials $b_0, b_1, \ldots, b_m$ as $(m+1)$-dimensional integral vectors (where the coefficient of X^i is identified with the $(i+1)$-th coordinate), we see that the vectors $b_0, b_1, \ldots, b_m \in \mathbb{Z}^{m+1}$ are linearly independent, because they form an upper-diagonal matrix. It follows that L is an $(m+1)$-dimensional lattice in $\mathbb{Z}^{m+1}$. So, we have constructed an $(m+1)$-dimensional lattice containing h_0. We will show that k can be chosen in such a way that all elements of L that are linearly independent of h_0, are more than $2^{m/2}$ times longer than h_0, so that h_0 can be computed by means of the basis reduction algorithm (cf. Section 4).

Because h_0 is a factor of f in $\mathbb{Z}[X]$, we find from (2.2) that

$$|h_0| \leq \binom{2m}{m}^{\frac{1}{2}} |f| = B. \tag{5.2}$$

Suppose that k is chosen such that

$$p^k > (2^{m/2} B^2)^m. \tag{5.3}$$

Let g be an arbitrary non-zero element of L that is linearly independent of h_0; we claim that $|g| > 2^{m/2} B$, so that g is more than $2^{m/2}$ times longer than h_0.

Because g is linearly independent of h_0, and because h_0 is irreducible in $\mathbb{Z}[X]$, we have that $\gcd(h_0, g) = 1$ in $\mathbb{Z}[X]$. This implies that the polynomials $h_0 X^i$ for $0 \leq i < \text{degree}(g)$ and $g X^j$ for $0 \leq j < \text{degree}(h_0)$, regarded as $(\text{degree}(g)+\text{degree}(h_0))$-dimensional vectors, are linearly independent. The resultant $R \in \mathbb{Z}$ of h_0 and g is defined as the determinant of the matrix M having these vectors as columns. It follows that $R \neq 0$; from Hadamard's inequality and $\text{degree}(g) \leq m$, we get

$$|R| \leq |h_0|^m |g|^m. \tag{5.4}$$

The polynomials h_0 and g, reduced modulo p^k, have h as a common divisor in $(\mathbb{Z}/p^k\mathbb{Z})[X]$ (remember that both are elements of L). Therefore, the columns of M cannot be linearly independent when regarded modulo p^k, so that R must be zero modulo p^k. Combined with $R \neq 0$, we conclude that $p^k \leq |R|$, which implies, with (5.4), (5.3), and (5.2), that $|g| > 2^{m/2} B$. This proves our claim.

One problem remains to be solved, namely to determine the correct value of the degree m of h_0. This is simply done by applying the above construction for $m = \ell, \ell+1, \ldots, n-1$ in succession, where k is chosen such that (5.3) holds with m replaced by $n-1$:

$$(5.5) \qquad p^k > \left(2^{(n-1)/2}\binom{2(n-1)}{n-1}^{\frac{1}{2}} |f|\right)^{n-1}.$$

It follows from the above reasoning that for values of m smaller than the degree of h_0, the lattice does not contain any sufficiently short vectors, so that the first short vector that we find must be equal to $\pm h_0$ (note that the lattice of dimension $m+1$ contains all elements of the lattices of dimensions $\leq m$). If no short vector is found at all, then apparently $\text{degree}(h_0) > n-1$, so that $h_0 = f$.

Let us consider the running time of the L^3-algorithm. First, we observe that the factor h modulo p^k, with k such that (5.5) holds, can still be found in polynomial time. This follows from the running time estimates for the application of Hensel's lemma. Next, we see from (5.5) and the definition of the lattice, that the maximal number of bits in the coordinates of the basis for L is $O(k \log p) = O(n^2 + n \log|f|)$. Combined with the running time of the basis reduction algorithm (cf. Section 4), this implies that the applications of this algorithm can be done in time polynomial in n and $\log|f|$. We conclude that h_0 can be determined in polynomial time, so that f can also be factored in polynomial time.

A more careful description of the algorithm and analysis of its running time lead to the following theorem (cf. [II: Theorem 3.6]):

Theorem. *A primitive polynomial* $f \in \mathbb{Z}[X]$ *of degree* n *can be completely factored in* $O(n^{12} + n^9(\log|f|)^3)$ *bit operations.*

(Using the result from [9], which was mentioned in Section 4, this can be improved to $O(n^{11} + n^8(\log|f|)^3)$.)

If we apply Odlyzko's empirical result about the running time of the basis reduction algorithm, we get an $O(n^7 + n^4(\log|f|)^3)$ bound for the factoring algorithm. This indicates that the L^3-algorithm will not be of great practical value. This point will be further discussed in Section 7. First, we will consider some generalizations of the L^3-algorithm to other polynomial factoring problems.

6. Generalizations of the L^3-algorithm

In the previous section we have seen that primitive univariate polynomials with integral coefficients can be factored in polynomial time by an algorithm that essentially works as follows:

(i) compute a sufficiently precise, irreducible p-adic factor,

(ii) construct a lattice, such that the corresponding irreducible true factor is very short in this lattice, and

(iii) determine this true factor by means of the basis reduction algorithm.

In [III; V; VII; IV] we have shown that the same scheme can be used to formulate polynomial-time algorithms for various other kinds of polynomial factoring problems:

- polynomials in one variable with coefficients in an algebraic number field (cf. [III]; see [12] for an algorithm using norms, a technique due to Kronecker);
- multivariate polynomials with integral coefficients (cf. [V]);
- multivariate polynomials with coefficients in an algebraic number field (cf. [VII]);
- multivariate polynomials with coefficients in a finite field (cf. [IV]).

The algorithms described in these papers are polynomial-time in the length of a dense encoding of the polynomial to be factored. That is, application of one of our algorithms to a polynomial f of degree n_i in the variable X_i, for $1 \le i \le t$, can be done in time polynomial in $\Pi_{i=1}^{t} n_i$ and the size of the coefficients of f.

It is well known that this is not a very realistic complexity measure for multivariate polynomials. Theoretically however, our algorithms compare favorably to the straightforward generalization of the Berlekamp-Hensel algorithm. For the latter nothing better can be proved than a bound that is exponential in each of the degrees n_i. To get a realistic complexity measure, the length of a sparse encoding of the input polynomial and its (output) factors has to be considered; Von zur Gathen has shown that in this case algorithms can be given that run in expected polynomial time [18].

This is certainly not the place to go into the numerous details of the generalizations of the L^3-algorithm; these can be found in the papers referred to above. Instead, let us describe some of the most important points of two of our generalizations, namely $f \in \mathbb{Z}[X_1, X_2]$ and $f \in \mathbb{F}_q[X_1, X_2]$.

First, we consider the case $f \in \mathbb{Z}[X_1,X_2]$; this case is treated in detail in [V]. Step (i) can be generalized as follows. Let p be a prime number and let s be an integer. Denote by s_1 the ideal generated by p and X_2-s. Because $f \bmod s_1$ is a polynomial in $(\mathbb{Z}/p\mathbb{Z})[X_1]$, we can find its factorization in $(\mathbb{Z}/p\mathbb{Z})[X_1]$ by means of Berlekamp's algorithm. If p and s are chosen such that some square-freeness conditions are satisfied, then we can apply a generalized version of Hensel's lemma to lift the factorization of $f \bmod s_1$ to a factorization of f modulo the ideal s_k generated by p^k and $(X_2-s)^{n_2+1}$, for any $k \in \mathbb{Z}_{>0}$ that we want. In this way we compute a sufficiently precise p-adic factor h of f. Denote by ℓ the degree in X_1 of h, and assume that h is monic with respect to the variable X_1. We now turn to step (ii).

As in the L^3-algorithm, assume that the true factor h_0 of f that corresponds to h has degree m_1 in X_1 (and of course degree at most n_2 in X_2). A basis for the set L of multiples of h modulo the ideal s_k, having degree $\leq m_1$ in X_1 and $\leq n_2$ in X_2, is given by the polynomials

$$\{p^k X_2^j X_1^i : 0 \leq j \leq n_2,\ 0 \leq i < \ell\}$$
$$\cup\ \{(h X_2^j \bmod s_k)\, X_1^{i-\ell} : 0 \leq j \leq n_2,\ \ell \leq i \leq m_1\}.$$

Regarding these polynomials as $((n_2+1)(m_1+1))$-dimensional integral vectors, we see that they are linearly independent, and that the set L is an $((n_2+1)(m_1+1))$-dimensional lattice.

As in the proof of Section 5, consider the resultant $R \in \mathbb{Z}[X_2]$ of h_0 and an arbitrary non-zero element g of L, for which $\gcd(h_0,g) = 1$ (so, $R \neq 0$). An upper bound for the length $|R|$ of the vector R, as a function of $|h_0|$ and $|g|$, can easily be derived. Using this bound and (2.2), we see that $(X_2-s)^{n_2+1}$ cannot divide R if s is chosen sufficiently large. But $R \bmod s_k$ must be zero, because h divides both $h_0 \bmod s_k$ and $g \bmod s_k$, so that the absolute value of the maximal coefficient of $R \bmod (X_2-s)^{n_2+1} \in \mathbb{Z}[X_2]$ must be at least p^k. This yields a lower bound for the length of the vector g as a function of p^k, and we conclude that we can get g as long as we want by choosing k sufficiently large. This means that h_0 can be determined by the basis reduction algorithm (notice that $|h_0|$ can be bounded from above by a result from [6], cf. Section 2).

The algorithm sketched here can easily be extended to more than two variables. The initial substitution $X_2 = s$ is then replaced by a substitution $X_2 = s_2$, $X_3 = s_3, \ldots, X_t = s_t$, where t is the number of variables. The details of this construction for a slightly more complicated case, namely with coefficients in an algebraic number field, are given in [VII].

Another way of generalizing the L^3-algorithm to $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ is described in [VI]. There we apply the idea of the L^3-algorithm (i.e. a true factor is a short vector in a lattice) to formulate a polynomial-time reduction from factoring in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ to factoring in $\mathbb{Z}[X_1]$. For $\mathbb{Z}[X_1, X_2]$ this reduction follows from the above algorithm by replacing the factor modulo p^k and $(X_2-s)^{n_2+1}$ by a factor modulo $(X_2-s)^{n_2+1}$ only.

Other polynomial-time algorithms for factoring in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ are given by Kaltofen in [8] and Chistov and Grigoryev in [3]. As in [VI], Kaltofen reduced the problem of factoring in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ in polynomial time to factoring in $\mathbb{Z}[X_1]$; his reduction is completely different from ours. Chistov and Grigoryev applied some of Kaltofen's ideas and developed yet another reduction, this time from factoring in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ to factoring in $\mathbb{Z}[X_1]$ and $(\mathbb{Z}/p\mathbb{Z})[X_1, X_2, \ldots, X_t]$ (for the latter problem see below). All these algorithms can, in some way or another, be extended to polynomials with coefficients in an algebraic number field.

We now come to a different kind of generalization of the L^3-algorithm, an algorithm to factor polynomials in $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$, where $\mathbb{F}_q$ is a finite field containing q elements (cf. [IV]). We will briefly discuss this algorithm for $f \in \mathbb{F}_q[X_1, X_2]$. (An algorithm very similar to the one described here was independently discovered by Chistov and Grigoryev [3].)

In fact, the algorithm follows immediately from the L^3-algorithm by replacing the ring of integers $\mathbb{Z}$ by the ring of polynomials $\mathbb{F}_q[X_2]$. Consequently, the prime number p is replaced by an irreducible polynomial $F \in \mathbb{F}_q[X_2]$, and the factor modulo p^k by a factor modulo the ideal generated by F^k. Instead of a lattice in $\mathbb{Z}^{m+1}$, we now get an $\mathbb{F}_q[X_2]$-module $L \subset \mathbb{F}_q[X_2]^{m+1}$. Of course, as norm for the elements of L, the Euclidean length does not make sense here. Therefore, for an element x of $\mathbb{F}_q[X_2]^{m+1}$, we define the norm as the maximal degree in X_2 of any of the coordinates of x; so, 'short' means 'small degree in X_2'. Using this norm one easily proves that the true factor corresponding to a factor modulo (F^k) is the shortest element of L if k is sufficiently large (the proof follows the lines of

the proof in Section 5, but is much simpler).

Obviously we cannot use the basis reduction algorithm. To determine the shortest vector in this case, we remark that the shortest vector problem in $L \subset \mathbb{F}_q[X_2]^{m+1}$ can be formulated as a system of linear equations over $\mathbb{F}_q$. This gives a polynomial-time solution of the shortest vector problem at hand (cf. [IV: 3]). This concludes our brief explanation of the algorithm for $f \in \mathbb{F}_q[X_1, X_2]$.

Generalization to more than two variables follows in a similar way as we have seen before, namely by means of substitutions $X_3 = X_2^{k_3}, X_4 = X_2^{k_4}, \ldots, X_t = X_2^{k_t}$, for sufficiently large integers k_i (this is quite different from the way in which Chistov and Grigoryev extend their method to more variables [3]). In [10] another polynomial-time algorithm for $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$-factoring is given.

A more general approach to the generalizations of the L^3-algorithm will be presented in [17].

7. Practical algorithms

The algorithms from the previous sections have a worst-case running time that is bounded by a polynomial function of the length of a dense encoding of the input polynomial. Apart from the fact that a dense encoding gives an unrealistic complexity measure, the polynomial bound on the running time also does not imply that the algorithms are practical. Although the algorithms will perform much better than is suggested by their running times as analyzed in [II; V; VI; III; VII], we cannot expect them to be fast. This follows from Odlyzko's empirical running time of the basis reduction algorithm and from the dimension and coordinate-size of the lattices to which the basis reduction algorithm is applied. For instance, to factor $f \in \mathbb{Z}[X_1, X_2, \ldots, X_t]$ of degree n_i in X_i, the running time will be proportional to

$$\Pi_{i=1}^{t} n_i^7 + (\Pi_{i=1}^{t} n_i^4)(\log|f|)^3$$

at least (the theoretical bound from [VI] is $O(n_1^{3t-3}(\Pi_{i=1}^{t} n_i^{12} + (\Pi_{i=1}^{t} n_i^9)(\log|f|)^3))$). Hence, even for reasonably small values of n_i the running time will become prohibitively long.

The question arises which algorithms should be used in practice. For polynomials in one variable with integral coefficients the Berlekamp-Hensel algorithm usually performs very well, and if it does not, which is quite unlikely, we can apply the L^3-algorithm. For multivariate polynomials with integral coefficients we have the generalizations of the Berlekamp-Hensel algorithm, as mentioned in Section 3 (cf. [19]). These algorithms apply a reduction from the $\mathbb{Z}[x_1, x_2, \ldots, x_t]$-factoring problem to $\mathbb{Z}[x_1]$-factoring, for which nothing better than an exponential-time bound can be proved. This reduction however, appears to be very fast in practice, and the resulting factoring algorithm can be strongly recommended.

The same is true for polynomials with coefficients in an algebraic number field; the exponential-time generalized Berlekamp-Hensel algorithm (cf. [20]) will perform better than the polynomial-time generalized L^3-algorithm. It looks as though these factoring algorithms that use lattices and the basis reduction algorithm have merely theoretical value. Fortunately, this is not the case, as we have shown in [I], where an algorithm is described to factor polynomials with coefficients in an algebraic number field. The algorithm is based on the Berlekamp-Hensel algorithm, and therefore not polynomial-time, but it is much faster than the methods from [20; 21], which are also of the Berlekamp-Hensel type. The reason of the speed of our algorithm is that, up to the search for the true factors, no computations have to be performed on algebraic numbers. Instead, all computations can be done in $\mathbb{Z}/p^k\mathbb{Z}$ for a suitably chosen prime power p^k. We will briefly explain this algorithm.

Suppose that the algebraic number field $\mathbb{Q}(\alpha)$ is given as the field of rational numbers $\mathbb{Q}$ extended by a root α of a prescribed *minimal polynomial* $F \in \mathbb{Z}[T]$, i.e. $\mathbb{Q}(\alpha) \simeq \mathbb{Q}[T]/(F)$. Let $f \in \mathbb{Q}(\alpha)[X]$ be the polynomial to be factored. In the older algorithms (cf. [20; 21]) one tries to find a prime number p such that $F \bmod p$ is irreducible in $(\mathbb{Z}/p\mathbb{Z})[T]$, and such that some other conditions are satisfied. If such a prime number can be found, then the Berlekamp-Hensel approach can immediately be generalized by observing that $(\mathbb{Z}/p\mathbb{Z})[T]/(F \bmod p)$ is a field. Several techniques are developed for the case that such a prime number cannot be found. In [20] the problem is reduced to the problem of factoring a polynomial of much higher degree in $\mathbb{Z}[X]$ (a technique that uses norms, as in [12]). In [21] the p-adic factorization of the minimal polynomial is used to compute several p-adic factorizations of f; the true factors of f are then determined by means of the Chinese remainder

algorithm combined with a combinatorial search. In both cases the algorithms from [20; 21] are rather slow due to the time consuming computations in the algebraic number field.

The algorithm from [1] proceeds as follows. First, a prime number p is chosen such that, among other conditions, $F \bmod p$ has a linear factor $H \bmod p$ in $(\mathbb{Z}/p\mathbb{Z})[T]$. The polynomial f reduced modulo p and $H \bmod p$ is, due to the linearity of $H \bmod p$, contained in $(\mathbb{Z}/p\mathbb{Z})[X]$, and can easily be factored by means of Berlekamp's algorithm. This factorization can be lifted to a factorization of f modulo p^k and $H \bmod p^k$ for a sufficiently large value of k by means of Hensel's lemma, where $H \bmod p^k$ is the lifted linear factor of $F \bmod p^k$ in $(\mathbb{Z}/p^k\mathbb{Z})[T]$.

To find the true factors of f in $\mathbb{Q}(\alpha)[X]$ we look, as usual, at combinations of the p-adic factors of f. These combinations however are polynomials in $(\mathbb{Z}/p^k\mathbb{Z})[X]$, so we must be able to reconstruct the coefficients in $\mathbb{Q}(\alpha)$ from their images modulo p^k and $H \bmod p^k$. Let x be such a coefficient of a true factor, and let $\tilde{x}$ be its image modulo p^k and $H \bmod p^k$, so $\tilde{x} \in \mathbb{Z}/p^k\mathbb{Z}$. For simplicity we will assume that $x \in \mathbb{Z}[\alpha] = \mathbb{Z}[T]/(F)$. We will show how x can be found given $\tilde{x}$. If we regard x and $\tilde{x}$ as (degree(F))-dimensional integral vectors, then they are congruent modulo the (degree(F))-dimensional integral lattice

$$L = \{p^k r + (H \bmod p^k)\, q : r,q \in \mathbb{Z}[\alpha],\ \text{degree}(r) = 0,\ \text{degree}(q) < \text{degree}(F)-1\}.$$

Because x is a coefficient of a true factor, the vector x is of bounded length. From the same proof as in Section 5 we conclude that we can choose k such that all non-zero elements of L are much longer than x. This implies that x is the shortest element of $\mathbb{Z}^{\text{degree}(F)}$ that is congruent to $\tilde{x}$ modulo L, so that x can be found by reducing $\tilde{x}$ modulo a sufficiently orthogonal basis for L. As mentioned in Section 4 such a basis for L can be computed by the basis reduction algorithm.

We conclude that, during the search for the true factors, all algebraic numbers can be reconstructed in a unique way by means of one application of the basis reduction algorithm. This is the only practical application of lattices and the basis reduction algorithm to factoring polynomials that we know of up till now.

References

I. A.K. Lenstra, *Lattices and factorization of polynomials over algebraic number fields*, Proceedings Eurocam '82, European computer algebra conference, LNCS 144, 32-39.

II. A.K. Lenstra, H.W. Lenstra, Jr., L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann. 261 (1982), 515-534.

III. A.K. Lenstra, *Factoring polynomials over algebraic number fields*, rep. IW 213/82, Mathematisch Centrum, Amsterdam; extended abstract in Proceedings Eurocal '83, European computer algebra conference, LNCS 162, 245-254.

IV. A.K. Lenstra, *Factoring multivariate polynomials over finite fields*, rep. IW 221/83, Mathematisch Centrum, Amsterdam; to appear in the special STOC issue of *Computer and system sciences*.

V. A.K. Lenstra, *Factoring multivariate integral polynomials*, Proceedings 10-th international colloquium on automata, languages and programming, LNCS 154, 458-465; to appear in the special ICALP issue of *Theoretical computer science*.

VI. A.K. Lenstra, *Factoring multivariate integral polynomials, II*, rep. IW 230/83, Mathematisch Centrum, Amsterdam.

VII. A.K. Lenstra, *Factoring multivariate polynomials over algebraic number fields*, rep. IW 233/83, Mathematisch Centrum, Amsterdam.

1. E.R. Berlekamp, *Factoring polynomials over large finite fields*, Math. Comp. 24 (1970), 713-735.

2. W.S. Brown, *The subresultant PRS algorithm*, ACM Transactions on mathematical software 4 (1978), 237-249.

3. A.L. Chistov, D.Y. Grigoryev, *Polynomial-time factoring of the multivariable polynomials over a global field*, LOMI preprint E-5-82, Leningrad 1982.

4. G.E. Collins, *Factoring univariate integral polynomials in polynomial average time*, Proceedings Eurosam '79, 317-329.

5. U. Dieter, *How to calculate shortest vectors in a lattice*, Math. Comp. 29 (1975), 827-833.

6. A.O. Gel'fond, *Transcendental and algebraic numbers*, Dover Publ., New York 1960.

7. E. Kaltofen, D.R. Musser, B.D. Saunders, *A generalized class of polynomials that are hard to factor*, Proceedings 1981 ACM symposium on symbolic and algebraic computation, 188-194.

8. E. Kaltofen, *On the complexity of factoring polynomials with integer coefficients*, Ph.D. thesis, Rensselaer Polytechnic Institute, August 1982.

9. E. Kaltofen, *On the complexity of finding short vectors in integer lattices*, Proceedings Eurocal '83, European computer algebra conference, LNCS 162, 236-244.

10. J. Von zur Gathen, E. Kaltofen, *Polynomial-time factorization of multivariate polynomials over finite fields*, Proceedings 10-th international colloquium on automata, languages and programming, LNCS 154, 250-263.

11. D.E. Knuth, *The art of computer programming, Vol. 2, Seminumerical algorithms*, Addison Wesley, Reading, second edition 1981.

12. S. Landau, *Factoring polynomials over algebraic number fields*, to appear.

13. H.W. Lenstra, Jr., *Integer programming with a fixed number of variables*, Math. Oper. Res. 8 (1983), 538-548.

14. M. Mignotte, *An inequality about factors of polynomials*, Math. Comp. 28 (1974), 1153-1157.

15. M. Pohst, *On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications*, Sigsam Bulletin Vol. 15, number 1 (1981), 37-44.

16. P. Van Emde Boas, *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*, Rep. Dep. Math. 81-04, Univ. of Amsterdam, April 1981.

17. J. Von zur Gathen, *Hensel and Newton methods in valuation rings*, Math. Comp., to appear.

18. J. Von zur Gathen, *Factoring sparse multivariate polynomials*, Proceedings 24-th annual symposium on foundations of computer science (1983), 172-179.

19. P.S. Wang, L.P. Rothschild, *Factoring multivariate polynomials over the integers*, Math. Comp. 29 (1975), 935-950.

20. P.S. Wang, *Factoring multivariate polynomials over algebraic number fields*, Math. Comp. 30 (1976), 324-336.

21. P.J. Weinberger, L.P. Rothschild, *Factoring polynomials over algebraic number fields*, ACM Transactions on mathematical software 2 (1976), 335-350.

22. D.Y.Y. Yun, *The Hensel-lemma in algebraic manipulation*, MIT, Cambridge 1974; reprint: Garland Publ. Co., New York 1980.

23. H. Zassenhaus, *On Hensel factorization, I*, J. Number Theory 1 (1969), 291-311.

Lattices and Factorization of Polynomials over Algebraic Number Fields

(Extended Abstract)

A.K. Lenstra
Mathematisch Centrum
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

1. Introduction and Notation.

We present a new algorithm to factorize polynomials over an algebraic number field. The algebraic number field is given as the field of rational numbers extended by a root of a prescribed minimal polynomial. Unlike other algorithms the efficiency of our method does not depend on the irreducibility of the minimal polynomial modulo some prime.

A brief outline of our algorithm is as follows. First, we factorize the polynomial to be factored over a large enough ring determined by a prime power p^k and an irreducible factor of the minimal polynomial modulo p^k. We then construct a lattice such that the coefficients of the factors over the algebraic number field are congruent, modulo this lattice, to the coefficients of the factors over the ring. Using a theorem stating that these coefficients in the algebraic number field are the shortest-length vectors with this property, we are able to compute them, if a sufficiently orthogonal basis of the lattice can be found.

That such a basis can be effectively constructed is a result of H.W. Lenstra [4], which is presented in Section 2, together with a number of elementary remarks about lattices. In Section 3 we prove a theorem giving a lower bound for the length of a polynomial having modulo p^k a non-trivial common divisor with an irreducible polynomial. As an application of this theorem we describe the new algorithm for factorization of polynomials over algebraic number fields in Section 4; we include some machine examples with timings. In Section 5 we make some final remarks on our new method, and we show that the theorem from Section 3 can also be used to formulate a new algorithm for factoring in $\mathbb{Z}[X]$.

Throughout this paper we make no distinction between vectors and polynomials; an m-dimensional vector $v = (v_0,\ldots,v_{m-1})^T$ corresponds to the polynomial $v(X) = \Sigma_{i=0}^{\underline{dv}} v_i X^i$, where $\underline{dv}$ denotes the degree of the polynomial v (here $\underline{dv} = -1$ if $v_i = 0$ for $i = 0,\ldots,m-1$, and $\underline{dv} = \max\{ i \mid v_i \neq 0 \}$ otherwise). Conversely a polynomial $v(X) = \Sigma_{i=0}^{\ell} v_i X^i$ corresponds to an m-dimensional vector $v = (v_0,\ldots,v_\ell,0,\ldots,0)^T$ for all $m > \ell$. If $v = (v_0,\ldots,v_{m-1})^T \in \mathbb{R}^m$, we denote by $[v]$ the vector $w = (w_0,\ldots,w_{m-1})^T \in \mathbb{Z}^m$, such that w_i is the integer nearest to v_i for $i = 0,\ldots,m-1$, and where halves are rounded upwards, e.g. $[0.5] = 1$. Furthermore we put $\|v\| = (\Sigma_{i=0}^{m-1} v_i^2)^{\frac{1}{2}}$, the *length* of v.

2. Lattices.

Let $b_0,\ldots,b_{m-1} \in \mathbb{Z}^m$ be m linearly independent vectors. The *lattice* L with basis $b_0,\ldots,b_{m-1}$ is defined as $L = \sum_{j=0}^{m-1} \mathbb{Z} b_j$. Putting $M = (b_0|\ldots|b_{m-1})$, the m×m matrix with b_j, $j = 0,\ldots,m-1$, as columns, we define the *determinant* of L as $d(L) = |\det((b_i,b_j)_{i,j=0}^{m-1})|^{\frac{1}{2}} = |\det(M)|$; the value of d(L) is independent of the choice of the basis of L. By the *fundamental domain* of $b_0,\ldots,b_{m-1}$ we denote the set $\{x \in \mathbb{R}^m \mid \exists c_j \in [-\frac{1}{2},\frac{1}{2}),\ j = 0,\ldots,m-1,$ such that $x = \sum_{j=0}^{m-1} c_j b_j\}$. For all $\tilde{x} \in \mathbb{R}^m$ there exists a unique element $x = \tilde{x} - M\cdot[M^{-1}\cdot\tilde{x}]$ in the fundamental domain, such that x and $\tilde{x}$ are congruent modulo L.

A measure of the orthogonality of a basis $b_0,\ldots,b_{m-1}$ is given by the *orthogonality defect* OD: $OD(b_0,\ldots,b_{m-1}) = \prod_{j=0}^{m-1} \|b_j\|/d(L)$. From Hadamard's inequality we know that $OD \geq 1$, but there is no a priori upper bound for OD. In [4] an algorithm is given to construct a basis for an arbitrary lattice such that the orthogonality defect of this basis is bounded from above by a constant depending on the dimension of the lattice only.

Theorem 1. (Reduction Algorithm) For any choice of $z \in (0,\frac{1}{2}\sqrt{3})$ we can reduce an arbitrary basis of an m-dimensional lattice L to a basis $b_0,\ldots,b_{m-1} \in \mathbb{Z}^m$ of L satisfying $1 \leq OD(b_0,\ldots,b_{m-1}) \leq (\frac{4z^2+1}{4z^2})^{m\cdot(m-1)/4}$. □

The running time of this algorithm is exponential in the dimension of the lattice; for small dimensions (i.e. ≤ 10) this appears to be no serious drawback. In the sequel we put $C = C(z,m) = (\frac{4z^2+1}{4z^2})^{m\cdot(m-1)/4}$. In practice the value for z doesn't matter too much; all our applications of Theorem 1 resulted in bases satisfying OD < 2 (which is however certainly not always possible).

It is intuitively clear that the radius of the largest sphere contained in the fundamental domain is proportional to 1/OD. The following lemma makes this more precise.

Lemma 1. Let $0 < B < \min_{0 \leq j < m} \|b_j\|$. The fundamental domain of $b_0,\ldots,b_{m-1}$ contains an m-dimensional sphere about the origin with radius > B/(2·OD), and all vectors ≠ 0 in L have length > B/OD. □

It follows that if all vectors ≠ 0 in an arbitrary lattice have length > B, we can construct a basis such that the fundamental domain of this basis contains a sphere about the origin with radius at least B/(2·C).

3. A lower bound theorem.

Let $F \in \mathbb{Z}[T]$ be an irreducible polynomial of degree m, and let $H_k \in (\mathbb{Z}/p^k\mathbb{Z})[T]$ be a monic factor of degree ℓ, $1 \leq \ell < m$, of F modulo p^k, for some prime power p^k. We define the m-dimensional lattice L_k generated by H_k and p^k as the lattice with the following basis: $b_i = p^k\cdot T^i$, $i = 0,\ldots,\ell-1$,

$b_i = H_k\cdot T^{i-\ell}$, $i = \ell,\ldots,m-1$.

Here the polynomials b_i are regarded as m-dimensional vectors. Clearly $b_0,\ldots,b_{m-1}$ are linearly independent and $d(L_k) = p^{k\cdot\ell}$. Remark that L_k equals the set of polynomials of degree < m having H_k as a factor modulo p^k.

We prove that for all $B > 0$ we can find an index $k_0 = k_0(B)$, such that the fundamental domain of the reduced basis of L_k contains a sphere with radius $> B$, for all $k \geq k_0$. We do this by proving that the lengths of the vectors $\neq 0$ in L_k can be bounded from below by a monotone increasing function of k.

Theorem 2. Let $V_k \neq 0$ be an arbitrary vector in L_k, regarded as polynomial in $\mathbb{Z}[T]$ of degree $n < m$, then $p^{k \cdot \ell} \leq \|F\|^n \cdot \|V_k\|^m \leq \|F\|^{m-1} \cdot \|V_k\|^m$.

Proof. Since F is irreducible over $\mathbb{Z}$ and $n < m$, we have that $\gcd(F, V_k) = 1$ over $\mathbb{Z}$, and therefore $G_1 \cdot F + G_2 \cdot V_k = 0$ if and only if $G_1 = G_2 = 0$, where $G_1, G_2 \in \mathbb{Z}[T]$ and $\underline{d}G_1 < n$, $\underline{d}G_2 < m$. This implies that the collection

$$\tilde{b}_i = F \cdot T^i, \quad i = 0, \ldots, n-1,$$
$$\tilde{b}_i = V_k \cdot T^{i-n}, \quad i = n, \ldots, n+m-1,$$

constitutes a basis of an $(n+m)$-dimensional lattice L contained in $\{\mathbb{Z} + \mathbb{Z} \cdot T + \ldots + \mathbb{Z} \cdot T^{n+m-1}\}$ with $d(L) \leq \|F\|^n \cdot \|V_k\|^m$ (Hadamard's inequality). The polynomials F and V_k both have the monic polynomial H_k as a factor modulo p^k, and therefore the lattice L is a sublattice of the $(n+m)$-dimensional lattice L_k' generated by H_k and p^k, so that

$$d(L_k') = p^{k \cdot \ell} \leq d(L) \leq \|F\|^n \cdot \|V_k\|^m. \quad \square$$

Remark that up to the constant factor, the lower bound $\|F\|^{-(m-1)/m} \cdot p^{(k \cdot \ell)/m}$ for elements in L_k is the best possible. This follows from Theorem 1, namely there exists a basis $b_0, \ldots, b_{m-1}$ of L_k such that $\prod_{j=0}^{m-1} \|b_j\| \leq C(z,m) \cdot p^{k \cdot \ell}$. Therefore there is a basisvector b_i satisfying $\|b_i\| \leq C(z,m)^{1/m} \cdot p^{(k \cdot \ell)/m}$.

It follows from Theorem 2, and from the results of the previous section, that in order to obtain a sphere with radius B, we should take k such that

$$\|F\|^{m-1} \cdot (2 \cdot C(z,m) \cdot B)^m < p^{k \cdot \ell}. \qquad (*)$$

We are now able to solve the following problem. Given a value $B > 0$ and a polynomial $\tilde{w} \in \mathbb{Z}[T]/H_k$ where k satisfies $(*)$, determine if possible a polynomial $w \in \mathbb{Z}[T]/F$ such that $\|w\| \leq B$ and such that $\tilde{w}$ and w are congruent modulo H_k and p^k. Clearly, if w exists then w is unique and $w = \tilde{w} - M \cdot [M^{-1} \cdot \tilde{w}]$, where M is the matrix of the reduced basis of L_k. Remark that if we have a number of polynomials $\tilde{w}$, we only have to compute M and (the first ℓ columns of) M^{-1} once.

4. Factorization in $(\mathbb{Q}(\alpha))[X]$.

We are ready to present our new algorithm for factoring polynomials over algebraic number fields. Let $\mathbb{Q}(\alpha)$ be an algebraic number field, where α denotes a zero of a monic irreducible polynomial F of degree m over $\mathbb{Z}$.

Lattice algorithm (LA).

Given a square-free monic polynomial $f \in (\mathbb{Q}(\alpha))[X]$ of degree n, this algorithm computes the irreducible factors of f over $\mathbb{Q}(\alpha)$.

1) Determine $D \in \mathbb{N}$, such that f and the factors of f are in $(\frac{1}{D}\mathbb{Z}[\alpha])[X]$.
2) Choose a prime p such that
   - $p \nmid D$,

- F remains square-free modulo p,
- F has a non-trivial monic irreducible factor H_1 of degree ℓ modulo p,
- f remains square-free modulo H_1 and p.

3) Choose B such that B/D is an upper bound for the length of the coefficients (in $\frac{1}{D}\mathbb{Z}[\alpha]$) of the factors of f over $\mathbb{Q}(\alpha)$.

4) Take $k \in \mathbb{N}$ minimal such that (*) holds, and determine the monic irreducible factor H_k of degree ℓ of F modulo p^k, such that $H_k \equiv H_1$ modulo p.

5) Determine the complete factorization of f modulo H_k and p^k:
$$(D^{-1} \bmod p^k)\cdot(D\cdot f) \equiv \Pi_{i=1}^{r} h_i \text{ modulo } (H_k, p^k).$$

6) If $r = 1$ then f is irreducible. Otherwise compute M, the matrix of the reduced basis of the m-dimensional lattice L_k generated by H_k and p^k. Compute the polynomial $\tilde{h} = ((D\cdot\Pi_{i\in S} h_i) \text{ modulo } (H_k, p^k)) = \Sigma_{i=0}^{d\tilde{h}} \tilde{v}_i X^i$ for all subsets $S \subset \{1,\ldots,r\}$ such that $\underline{d}\tilde{h} \le \lfloor n/2 \rfloor$, and test whether
$$h = \frac{1}{D}\cdot(\Sigma_{i=0}^{dh}(\tilde{v}_i - M\cdot[M^{-1}\cdot\tilde{v}_i])X^i) \in (\tfrac{1}{D}\mathbb{Z}[\alpha])[X]$$
is a factor of f over $\frac{1}{D}\mathbb{Z}[\alpha]$.

The values of D and B in Steps 1) and 3) can be determined using methods from [9]. The theoretical value for B is often much too large; it is in general advisable to use a heuristic bound [3,7]. The factorization of f modulo H_k and p^k is computed in the usual way; first factorize f modulo H_1 and p using for instance the Cantor-Zassenhaus algorithm [1] for factorization over finite fields (after Step 2), exit if $r = 1$), next apply Zassenhaus' quadratic lift-algorithm [10,11] to obtain the factors modulo H_k and p^k. It follows from Section 3 that the fundamental domain of the reduced basis of L_k contains the coefficients of the factors of f over $\mathbb{Q}(\alpha)$ (multiplied by D). These factors can therefore be determined as described in Step 6). Remark that all integers occurring in the LA are in absolute value $< p^{2k}$.

In practice we replace C(z,m) in (*) by 2, thus obtaining a smaller value for k. If the orthogonality defect of the reduced basis of L_k turns out to be too large (i.e. $\min_{0\le j<m} \|b_j\|/(2\cdot OD(b_0,\ldots,b_{m-1})) < B$) we try again with a larger k, but in most cases OD will be small enough.

As an example we factorize a polynomial from Weinberger and Rothschild [9] using the LA. Let $F(T) = T^6+3T^5+6T^4+T^3-3T^2+12T+16$ ($m = 6$), and let $f = X^3-3 \in (\mathbb{Q}(\alpha))[X]$ ($n = 3$), where α denotes a zero of F.

1) Like Weinberger and Rothschild we use $D = 12$ as the denominator of the factors of f over $\mathbb{Q}(\alpha)$.

2) The prime $p = 7$ satisfies the conditions; we find $H_1 = T^3+T^2-2T+3$ and $\ell = 3$.

3) We know from Weinberger and Rothschild that 40/12 is an upper bound for the length of the coefficients of the factors of f, so we take $B = 40$.

4) We replace C(z,6) in (*) by 2 and we take k minimal such that $(\sqrt{456})^5\cdot(2\cdot2\cdot40)^6 < 7^{k\cdot3}$, so we find $k = 8$, and $H_8 = T^3-1399040T^2-1399043T-4$.

5) $f \equiv (X-2387947\alpha-2387948)\cdot(X+2387948\alpha+1)\cdot(X-\alpha+2387947)$ modulo $(\alpha^3-1399040\alpha^2-1399043\alpha-4,\ 7^8)$.

6) Application of Theorem 1 to L_8 yields the following matrix:

$$\begin{pmatrix} 1265 & -1265 & -1059 & -1265 & 0 & -103 \\ 479 & -273 & -547 & 683 & 2530 & 34 \\ 547 & 547 & -137 & -34 & 752 & 1641 \\ -752 & -2017 & 2359 & -752 & 0 & -171 \\ -957 & -205 & -957 & -1231 & 1265 & 205 \\ -1299 & -1299 & -376 & 2051 & -752 & 376 \end{pmatrix} = M.$$

The orthogonality defect of this basis is $(\Pi_{i=0}^{5}\|b_i\|)/7^{8\cdot 3} = 1.4 < 2$, so k is large enough. Remark that according to Lemma 1 the radius of the sphere contained in the fundamental domain of this basis is at least $\lfloor \min_{i=0,\ldots,5} \|b_i\|/(2\cdot OD) \rfloor > 600$.

The highest power of α in the above factorization of f is one, so we have to compute only the first two columns of the inverse of M:

$$\begin{pmatrix} 2.5500\ 10^{-4} & 1.3045\ 10^{-4} & * & * & * & * \\ -2.8466\ 10^{-4} & -0.7112\ 10^{-4} & * & * & * & * \\ -1.8977\ 10^{-4} & 0.2966\ 10^{-4} & * & * & * & * \\ -0.9489\ 10^{-4} & 1.8977\ 10^{-4} & * & * & * & * \\ -0.9489\ 10^{-4} & 3.2022\ 10^{-4} & * & * & * & * \\ 0.3556\ 10^{-4} & -1.6011\ 10^{-4} & * & * & * & * \end{pmatrix} = M^{-1}.$$

First we take $S = \{1\}$: $\tilde{h} = (12\cdot(X-2387947\alpha-2387948)) \bmod (H_8, 7^8) = 12X+168641\alpha+168629 = \tilde{v}_1 X+\tilde{v}_0$. Now reduce these coefficients modulo the reduced basis of L_8: $h = \frac{1}{12}\cdot\Sigma_{i=0}^{1}(\tilde{v}_i - M\cdot[M^{-1}\cdot\tilde{v}_i])X^i = X-(\alpha^5+3\alpha^4+6\alpha^3+5\alpha^2-3\alpha+12)/12$, and indeed h is a factor of f over $\mathbb{Q}(\alpha)$. For $S = \{2\}$ we find the factor $X+(\alpha^5+2\alpha^4+4\alpha^3-\alpha^2+4\alpha+14)/6$, so that the complete factorization of f over $\mathbb{Q}(\alpha)$ becomes $f = (X-(\alpha^5+3\alpha^4+6\alpha^3+5\alpha^2-3\alpha+12)/12)\cdot(X+(\alpha^5+2\alpha^4+4\alpha^3-\alpha^2+4\alpha+14)/6)\cdot(X-(\alpha^5+\alpha^4+2\alpha^3-7\alpha^2+11\alpha+16)/12)$.

We implemented the LA and the algorithm as described by Weinberger and Rothschild [9] (WRA) in Algol 68 on a CDC-Cyber 170-750 computer (we didn't implement the methods described in [5,6,7]). Below we give a number of machine examples; we denote by "new time" and "old time" the time taken by the LA and the time taken by the WRA respectively (in milliseconds).

1) $f = \frac{1}{47}(47X^6+21X^5+598X^4+1561X^3+1198X^2+261X+47)$, $\alpha^2-\alpha+3 = 0$.

$\alpha-1 \equiv 0$ modulo 3: new time 143 msec,

irreducible modulo 7: old time 676 msec.

factorization over $\mathbb{Q}(\alpha)$:

$\frac{1}{2209}(47X^3-(121\alpha-71)X^2-(121\alpha+70)X-47)\cdot(47X^3+(121\alpha-50)X^2+(121\alpha-191)X-47)$.

2) $f = \frac{1}{16}(16X^6-1)$, $\alpha^3+2 = 0$.

$\alpha^2+2\alpha-1 \equiv 0$ modulo 5: new time 431 msec,

irreducible modulo 7: old time 511 msec.

factorization over $\mathbb{Q}(\alpha)$:

$\frac{1}{64}(4X^2+2\alpha X+\alpha^2)\cdot(4X^2-2\alpha X+\alpha^2)\cdot(2X-\alpha)\cdot(2X+\alpha)$.

3) $f = X^8-X^7-X^6+X^4-X^2+X+1$, $\alpha^4-\alpha+1 = 0$.

$\alpha^3-\alpha^2+\alpha+1 \equiv 0$ modulo 3: new time 1347 msec,

$\alpha+1 \equiv 0$ modulo 3: new time 235 msec,

irreducible modulo 7: old time 2038 msec.

factorization over $\mathbb{Q}(\alpha)$:
$(X^6-(\alpha^3+\alpha^2+\alpha)X^5+(2\alpha^3+\alpha^2-3)X^4+(\alpha^3+2\alpha^2+2\alpha)X^3-(2\alpha^3+\alpha^2-3)X^2-(\alpha^3+\alpha^2+\alpha)X-1)\cdot(X^2+(\alpha^3+\alpha^2+\alpha-1)X-1)$.

4) $f=X^3-3$, $\alpha^6+3\alpha^5+6\alpha^4+\alpha^3-3\alpha^2+12\alpha+16=0$.
$\alpha^2-2\alpha-1\equiv 0$ modulo 5: new time 564 msec,
two factors modulo 7: old time 814 msec.
factorization over $\mathbb{Q}(\alpha)$:
$\frac{1}{864}(12X-\alpha^5-3\alpha^4-6\alpha^3-5\alpha^2+3\alpha-12)\cdot(6X+\alpha^5+2\alpha^4+4\alpha^3-\alpha^2+4\alpha+14)\cdot(12X-\alpha^5-\alpha^4-2\alpha^3+7\alpha^2-11\alpha-16)$.

5) $f=X^9+9X^8+36X^7+69X^6+36X^5-99X^4-303X^3-450X^2-342X-226$, $\alpha^9-15\alpha^6-87\alpha^3-125=0$.
$\alpha^3-\alpha+2\equiv 0$ modulo 7: new time 2816 msec,
three factors modulo 7: old time 59183 msec.
factorization over $\mathbb{Q}(\alpha)$:
$(X^6+6X^5+15X^4+(\alpha^3+5)X^3+(3\alpha^3-30)X^2+(3\alpha^3-39)X+\alpha^6-14\alpha^3-101)\cdot(X^2+(\alpha+2)X+\alpha^2+\alpha+1)\cdot(X-\alpha+1)$.

5. Remarks.

From the examples in the previous section we conclude that, as we expected, the use of the LA can be recommended, as long as the degree of the minimal polynomial is not too large. Even in the case that the minimal polynomial remains irreducible modulo some small prime the LA is considerably faster than the WRA.

A drawback of the LA is the rather large theoretical lower bound for p^k. This causes no difficulties in an implementation using arbitrary length integers, but in the case that fixed length integers are used (as in our implementation, where we used single-length integers of 48 bits) we can get problems. There are several ways to lower the value for p^k if the theoretical bound on p^k appears to be too large.

1) Don't care about the theoretical bound, take p^k as large as the implementation allows. If the reduced basis $b_0,\ldots,b_{m-1}$ satisfies $\min_{0\le j<m}\|b_j\|/(2\cdot OD(b_0,\ldots,b_{m-1}))>B$ then the complete factorization will be found. Otherwise just try to find factors, but no guarantee can be given that we find them all.
2) Try to find a large degree irreducible factor of the minimal polynomial.
3) Use a combination of the WRA and the LA, i.e. combine the factorizations of f modulo a number of irreducible factors of the minimal polynomial modulo p^k (WRA), and apply the LA to these combinations. Here the lattice is generated by the product of this number of factors of the minimal polynomial and p^k. The running time of this algorithm grows exponentially with the number of factors of the minimal polynomial used, but unlike the WRA we do not have to use the complete factorization of the minimal polynomial; just take a number of factors such that the sum of the degrees is large enough to lower p^k sufficiently.
4) Any combination of 1), 2) and 3).

Theorem 2 can also be used while factoring in $\mathbb{Z}[X]$ [8]. Let $G\in\mathbb{Z}[X]$, and let H_k be a monic irreducible factor of degree ℓ of G modulo p^k. We test whether H_k leads

to an irreducible factor F of degree m of G (i.e. $H_k | F$ modulo p^k and $F | G$ over $\mathbb{Z}$) by looking at the (m+1)-dimensional lattice L_k generated by H_k and p^k. A basis of this lattice is given by: $b_i = p^k \cdot X^i$, $i = 0,\ldots,\ell-1$, $b_i = H_k \cdot X^{i-\ell}$, $i = \ell,\ldots,m$. If F exists then clearly $F \in L_k$, but also F is the shortest-length vector in L_k if k is chosen sufficiently large. This follows from a generalized version of Theorem 2, stating that if $V_k \in L_k$ such that $\gcd(F,V_k) = 1$ over $\mathbb{Z}$, then $p^{k\cdot\ell} \leq \|F\|^{\underline{d}V_k} \cdot \|V_k\|^m$. We know that there exists an effectively computable bound $B > 0$ such that $\|F\| < B$, so if we take k minimal such that $B^{2\cdot m} < p^{k\cdot\ell}$, then $B^{2\cdot m} < \|F\|^{\underline{d}V_k} \cdot \|V_k\|^m \leq B^m \cdot \|V_k\|^m$. This implies $\|V_k\| > B$, which proves that indeed F is the shortest-length vector in L_k. Using for instance the shortest-vector algorithm of Dieter [2] we can determine F. It is not difficult to see that Theorem 1 can also be used to calculate F, if we take k such that $B^{2\cdot m} \cdot C(z,m+1)^m < p^{k\cdot\ell}$.

A similar algorithm, using the computation of a shortest vector in a lattice, can be applied to factorize in $(\mathbb{Q}(\alpha))[X]$. Determination of a monic factor of degree n leads to a lattice of dimension $n\cdot m+1$, where m is the degree of the minimal polynomial. As the shortest-vector algorithms are only efficient for small-dimensional lattices this is in general not a very practical method.

In Section 4 we have restricted ourselves to univariate polynomials; remark that the LA equally well applies to the multivariate case.

References.

1. D.G. Cantor & H. Zassenhaus, A New Algorithm for Factoring Polynomials Over Finite Fields, Math. Comp. 36 (1981), pp 587-592.
2. U. Dieter, How to calculate Shortest Vectors in a Lattice, Math. Comp. 29 (1975), pp 827-833.
3. A.K. Lenstra, Lattices and Factorization of Polynomials, Mathematisch Centrum, Amsterdam, Report IW 190/81.
4. H.W. Lenstra Jr., Integer programming with a fixed number of variables, University of Amsterdam, Department of Mathematics, Report 81-03.
5. B.M. Trager, Algebraic Factoring and Rational Function Integration, Proc. SYMSAC 76, pp 219-226.
6. B.L. van der Waerden, Moderne Algebra, Springer, Berlin, 1931.
7. P.S. Wang, Factoring Multivariate Polynomials over Algebraic Number Fields, Math. Comp. 30 (1976), pp 324-336.
8. P.S. Wang, An Improved Multivariate Polynomial Factoring Algorithm, Math. Comp. 32 (1978), pp 1215-1231.
9. P.J. Weinberger & L.P. Rothschild, Factoring Polynomials over Algebraic Number Fields, ACM Transactions on Math. Software 2 (1976), pp 335-350.
10. H. Zassenhaus, On Hensel Factorization, I, J. of Number Theory 1 (1969), pp 291-311.
11. H. Zassenhaus, A Remark on the Hensel Factorization Method, Math. Comp. 32 (1978), pp 287-292.

Addendum.

Recently L. Lovász invented a polynomial time reduction algorithm. Among others, this new reduction algorithm leads to a polynomial time algorithm for factoring polynomials with rational coefficients (see Section 5). A report describing the new polynomial factorization algorithm in detail is available from the Mathematisch Centrum, Amsterdam.

A.K. Lenstra, H.W. Lenstra & L. Lovász, Factoring Polynomials with Rational Coefficients, Mathematisch Centrum, Amsterdam.

Math. Ann. 261, 515–534 (1982)

Mathematische Annalen

Factoring Polynomials with Rational Coefficients

A. K. Lenstra[1], H. W. Lenstra, Jr.[2], and L. Lovász[3]

1 Mathematisch Centrum, Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands
2 Mathematisch Instituut, Universiteit van Amsterdam, Roetersstraat 15, NL-1018 WB Amsterdam, The Netherlands
3 Bolyai Institute, A. József University, Aradi vértanúk tere 1, H-6720 Szeged, Hungary

In this paper we present a polynomial-time algorithm to solve the following problem: given a non-zero polynomial $f \in \mathbb{Q}[X]$ in one variable with rational coefficients, find the decomposition of f into irreducible factors in $\mathbb{Q}[X]$. It is well known that this is equivalent to factoring *primitive* polynomials $f \in \mathbb{Z}[X]$ into irreducible factors in $\mathbb{Z}[X]$. Here we call $f \in \mathbb{Z}[X]$ primitive if the greatest common divisor of its coefficients (the *content* of f) is 1.

Our algorithm performs well in practice, cf. [8]. Its running time, measured in bit operations, is $O(n^{12}+n^9(\log|f|)^3)$. Here $f \in \mathbb{Z}[X]$ is the polynomial to be factored, $n=\deg(f)$ is the degree of f, and

$$\left|\sum_i a_i X^i\right| = \left(\sum_i a_i^2\right)^{1/2}$$

for a polynomial $\sum_i a_i X^i$ with real coefficients a_i.

An outline of the algorithm is as follows. First we find, for a suitable small prime number p, a p-adic irreducible factor h of f, to a certain precision. This is done with Berlekamp's algorithm for factoring polynomials over small finite fields, combined with Hensel's lemma. Next we look for the irreducible factor h_0 of f in $\mathbb{Z}[X]$ that is divisible by h. The condition that h_0 is divisible by h means that h_0 belongs to a certain lattice, and the condition that h_0 divides f implies that the coefficients of h_0 are relatively small. It follows that we must look for a "small" element in that lattice, and this is done by means of a basis reduction algorithm. It turns out that this enables us to determine h_0. The algorithm is repeated until all irreducible factors of f have been found.

The basis reduction algorithm that we employ is new, and it is described and analysed in Sect. 1. It improves the algorithm given in a preliminary version of [9, Sect. 3]. At the end of Sect. 1 we briefly mention two applications of the new algorithm to diophantine approximation.

The connection between factors of f and reduced bases of a lattice is treated in detail in Sect. 2. The theory presented here extends a result appearing in [8, Theorem 2]. It should be remarked that the latter result, which is simpler to prove, would in principle have sufficed for our purpose.

0025-5831/82/0261/0515/$04.00

Section 3, finally, contains the description and the analysis of our algorithm for factoring polynomials.

It may be expected that other irreducibility tests and factoring methods that depend on diophantine approximation (Cantor [3], Ferguson and Forcade [5], Brentjes [2, Sect. 4A], and Zassenhaus [16]) can also be made into polynomial-time algorithms with the help of the basis reduction algorithm presented in Sect. 1.

Splitting an arbitrary non-zero polynomial $f \in \mathbb{Z}[X]$ into its *content* and its *primitive part*, we deduce from our main result that the problem of factoring such a polynomial is polynomial-time reducible to the problem of factoring positive integers. The same fact was proved by Adleman and Odlyzko [1] under the assumption of several deep and unproved hypotheses from number theory.

The generalization of our result to algebraic number fields and to polynomials in several variables is the subject of future publications.

1. Reduced Bases for Lattices

Let n be a positive integer. A subset L of the n-dimensional real vector space $\mathbb{R}^n$ is called a *lattice* if there exists a basis $b_1, b_2, \ldots, b_n$ of $\mathbb{R}^n$ such that

$$L = \sum_{i=1}^{n} \mathbb{Z} b_i = \left\{ \sum_{i=1}^{n} r_i b_i : r_i \in \mathbb{Z} (1 \leqq i \leqq n) \right\}.$$

In this situation we say that $b_1, b_2, \ldots, b_n$ form a *basis* for L, or that they *span* L. We call n the *rank* of L. The *determinant* $d(L)$ of L is defined by

$$d(L) = |\det(b_1, b_2, \ldots, b_n)|, \tag{1.1}$$

the b_i being written as column vectors. This is a positive real number that does not depend on the choice of the basis [4, Sect. I.2].

Let $b_1, b_2, \ldots, b_n \in \mathbb{R}^n$ be linearly independent. We recall the Gram-Schmidt orthogonalization process. The vectors b_i^* $(1 \leqq i \leqq n)$ and the real numbers μ_{ij} $(1 \leqq j < i \leqq n)$ are inductively defined by

$$b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^*, \tag{1.2}$$

$$\mu_{ij} = (b_i, b_j^*)/(b_j^*, b_j^*), \tag{1.3}$$

where $(,)$ denotes the ordinary inner product on $\mathbb{R}^n$. Notice that b_i^* is the projection of b_i on the orthogonal complement of $\sum_{j=1}^{i-1} \mathbb{R} b_j$, and that $\sum_{j=1}^{i-1} \mathbb{R} b_j = \sum_{j=1}^{i-1} \mathbb{R} b_j^*$, for $1 \leqq i \leqq n$. It follows that $b_1^*, b_2^*, \ldots, b_n^*$ is an orthogonal basis of $\mathbb{R}^n$.

In this paper, we call a basis $b_1, b_2, \ldots, b_n$ for a lattice L *reduced* if

$$|\mu_{ij}| \leqq 1/2 \quad \text{for} \quad 1 \leqq j < i \leqq n \tag{1.4}$$

and

$$|b_i^* + \mu_{i\,i-1} b_{i-1}^*|^2 \geqq \tfrac{3}{4} |b_{i-1}^*|^2 \quad \text{for} \quad 1 < i \leqq n, \tag{1.5}$$

where $||$ denotes the ordinary Euclidean length. Notice that the vectors $b_i^* + \mu_{i\,i-1} b_{i-1}^*$ and b_{i-1}^* appearing in (1.5) are the projections of b_i and b_{i-1} on the orthogonal complement of $\sum_{j=1}^{i-2} \mathbb{R} b_j$. The constant $\frac{3}{4}$ in (1.5) is arbitrarily chosen, and may be replaced by any fixed real number y with $\frac{1}{4} < y < 1$.

(1.6) **Proposition.** *Let $b_1, b_2, \ldots, b_n$ be a reduced basis for a lattice L in $\mathbb{R}^n$, and let $b_1^*, b_2^*, \ldots, b_n^*$ be defined as above. Then we have*

$$|b_j|^2 \leqq 2^{i-1} \cdot |b_i^*|^2 \quad \text{for} \quad 1 \leqq j \leqq i \leqq n, \tag{1.7}$$

$$d(L) \leqq \prod_{i=1}^{n} |b_i| \leqq 2^{n(n-1)/4} \cdot d(L), \tag{1.8}$$

$$|b_1| \leqq 2^{(n-1)/4} \cdot d(L)^{1/n}. \tag{1.9}$$

Remark. If $\frac{3}{4}$ in (1.5) is replaced by y, with $\frac{1}{4} < y < 1$, then the powers of 2 appearing in (1.7), (1.8) and (1.9) must be replaced by the same powers of $4/(4y-1)$.

Remark. From (1.8) we see that a reduced basis is also reduced in the sense of [9, (7)].

Proof of (1.6). From (1.5) and (1.4) we see that

$$|b_i^*|^2 \geqq (\tfrac{3}{4} - \mu_{i\,i-1}^2) \cdot |b_{i-1}^*|^2 \geqq \tfrac{1}{2} \cdot |b_{i-1}^*|^2$$

for $1 < i \leqq n$, so by induction

$$|b_j^*|^2 \leqq 2^{i-j} \cdot |b_i^*|^2 \quad \text{for} \quad 1 \leqq j \leqq i \leqq n.$$

From (1.2) and (1.4) we now obtain

$$\begin{aligned} |b_i|^2 &= |b_i^*|^2 + \sum_{j=1}^{i-1} \mu_{ij}^2 |b_j^*|^2 \\ &\leqq |b_i^*|^2 + \sum_{j=1}^{i-1} \tfrac{1}{4} 2^{i-j} |b_i^*|^2 \\ &= (1 + \tfrac{1}{4}(2^i - 2)) \cdot |b_i^*|^2 \\ &\leqq 2^{i-1} \cdot |b_i^*|^2. \end{aligned}$$

It follows that

$$|b_j|^2 \leqq 2^{j-1} \cdot |b_j^*|^2 \leqq 2^{i-1} \cdot |b_i^*|^2$$

for $1 \leqq j \leqq i \leqq n$. This proves (1.7).

From (1.1), (1.2) it follows that

$$d(L) = |\det(b_1^*, b_2^*, \ldots, b_n^*)|$$

and therefore, since the b_i^* are pairwise orthogonal

$$d(L) = \prod_{i=1}^{n} |b_i^*|.$$

From $|b_i^*| \leqq |b_i|$ and $|b_i| \leqq 2^{(i-1)/2} \cdot |b_i^*|$ we now obtain (1.8). Putting $j = 1$ in (1.7) and taking the product over $i = 1, 2, \ldots, n$ we find (1.9). This proves (1.6).

Remark. Notice that the proof of the inequality

$$d(L) \leqq \prod_{i=1}^{n} |b_i| \tag{1.10}$$

did not require the basis to be reduced. This is *Hadamard's inequality.*

(1.11) **Proposition.** *Let $L \subset \mathbb{R}^n$ be a lattice with reduced basis $b_1, b_2, \ldots, b_n$. Then*

$$|b_1|^2 \leqq 2^{n-1} \cdot |x|^2$$

for every $x \in L$, $x \neq 0$.

Proof. Write $x = \sum_{i=1}^{n} r_i b_i = \sum_{i=1}^{n} r_i' b_i^*$ with $r_i \in \mathbb{Z}$, $r_i' \in \mathbb{R}$ $(1 \leqq i \leqq n)$. If i is the largest index with $r_i \neq 0$ then $r_i' = r_i$, so

$$|x|^2 \geqq r_i'^2 \cdot |b_i^*|^2 \geqq |b_i^*|^2 .$$

By (1.7), we have $|b_1|^2 \leqq 2^{i-1} \cdot |b_i^*|^2 \leqq 2^{n-1} \cdot |b_i^*|^2$. This proves (1.11).

(1.12) **Proposition.** *Let $L \subset \mathbb{R}^n$ be a lattice with reduced basis $b_1, b_2, \ldots, b_n$. Let $x_1, x_2, \ldots, x_t \in L$ be linearly independent. Then we have*

$$|b_j|^2 \leqq 2^{n-1} \cdot \max\{|x_1|^2, |x_2|^2, \ldots, |x_t|^2\}$$

for $j = 1, 2, \ldots, t$.

Proof. Write $x_j = \sum_{i=1}^{n} r_{ij} b_i$ with $r_{ij} \in \mathbb{Z}$ $(1 \leqq i \leqq n)$ for $1 \leqq j \leqq t$. For fixed j, let $i(j)$ denote the largest i for which $r_{ij} \neq 0$. Then we have, by the proof of (1.11)

$$|x_j|^2 \geqq |b_{i(j)}^*|^2 \tag{1.13}$$

for $1 \leqq j \leqq t$. Renumber the x_j such that $i(1) \leqq i(2) \leqq \ldots \leqq i(t)$. We claim that $j \leqq i(j)$ for $1 \leqq j \leqq t$. If not, then $x_1, x_2, \ldots, x_j$ would all belong to $\mathbb{R}b_1 + Rb_2 + \ldots + \mathbb{R}b_{j-1}$, a contradiction with the linear independence of $x_1, x_2, \ldots, x_t$. From $j \leqq i(j)$ and (1.7) we obtain, using (1.13):

$$|b_j|^2 \leqq 2^{i(j)-1} \cdot |b_{i(j)}^*|^2 \leqq 2^{n-1} \cdot |b_{i(j)}^*|^2 \leqq 2^{n-1} \cdot |x_j|^2$$

for $j = 1, 2, \ldots, t$. This proves (1.12).

Remark. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ denote the successive minima of $|\ |^2$ on L, see [4, Chap. VIII], and let $b_1, b_2, \ldots, b_n$ be a reduced basis for L. Then (1.7) and (1.12) easily imply that

$$2^{1-i} \lambda_i \leqq |b_i|^2 \leqq 2^{n-1} \lambda_i \quad \text{for} \quad 1 \leqq i \leqq n,$$

so $|b_i|^2$ is a reasonable approximation of λ_i.

(1.14) *Remark.* Notice that the number 2^{n-1} may in (1.11) be replaced by $\max\{|b_1|^2/|b_i^*|^2 : 1 \leqq i \leqq n\}$ and in (1.12) by $\max\{|b_j|^2/|b_i^*|^2 : 1 \leqq j \leqq i \leqq n\}$.

(1.15) We shall now describe an algorithm that transforms a given basis $b_1, b_2, \ldots, b_n$ for a lattice L into a reduced one. The algorithm improves the

algorithm given in a preliminary version of [9, Sect. 3]. Our description incorporates an additional improvement due to J. J. M. Cuppen, reducing our running time estimates by a factor n.

To initialize the algorithm we compute b_i^* $(1 \leqq i \leqq n)$ and μ_{ij} $(1 \leqq j < i \leqq n)$ using (1.2) and (1.3). In the course of the algorithm the vectors $b_1, b_2, \ldots, b_n$ will be changed several times, but always in such a way that they form a basis for L. After every change of the b_i we shall update the b_i^* and μ_{ij} in such a way that (1.2) and (1.3) remain valid.

At each step of the algorithm we shall have a current subscript $k \in \{1, 2, \ldots, n+1\}$. We begin with $k=2$.

We shall now iterate a sequence of steps that starts from, and returns to, a situation in which the following conditions are satisfied:

$$|\mu_{ij}| \leqq \tfrac{1}{2} \quad \text{for} \quad 1 \leqq j < i < k, \tag{1.16}$$

$$|b_i^* + \mu_{i\,i-1} b_{i-1}^*|^2 \geqq \tfrac{3}{4} |b_{i-1}^*|^2 \quad \text{for} \quad 1 < i < k. \tag{1.17}$$

These conditions are trivially satisfied if $k=2$.

In the above situation one proceeds as follows. If $k=n+1$ then the basis is reduced, and the algorithm terminates. Suppose now that $k \leqq n$. Then we first achieve that

$$|\mu_{k\,k-1}| \leqq \tfrac{1}{2} \quad \text{if} \quad k > 1. \tag{1.18}$$

If this does not hold, let r be the integer nearest to $\mu_{k\,k-1}$, and replace b_k by $b_k - r b_{k-1}$. The numbers μ_{kj} with $j < k-1$ are then replaced by $\mu_{kj} - r\mu_{k-1\,j}$, and $\mu_{k\,k-1}$ by $\mu_{k\,k-1} - r$. The other μ_{ij} and all b_i^* are unchanged. After this change (1.18) holds.

Next we distinguish two cases.

Case 1. Suppose that $k \geqq 2$ and

$$|b_k^* + \mu_{k\,k-1} b_{k-1}^*|^2 < \tfrac{3}{4} |b_{k-1}^*|^2. \tag{1.19}$$

Then we interchange b_{k-1} and b_k, and we leave the other b_i unchanged. The vectors b_{k-1}^* and b_k^* and the numbers $\mu_{k\,k-1}$, $\mu_{k-1\,j}$, μ_{kj}, $\mu_{i\,k-1}$, μ_{ik}, for $j < k-1$ and for $i > k$, have now to be replaced. This is done by formulae that we give below. The most important one of these changes is that b_{k-1}^* is replaced by $b_k^* + \mu_{k\,k-1} b_{k-1}^*$; so the new value of $|b_{k-1}^*|^2$ is less than $\frac{3}{4}$ times the old one. These changes being made, we replace k by $k-1$. Then we are in the situation described by (1.16) and (1.17), and we proceed with the algorithm from there.

Case 2. Suppose that $k=1$ or

$$|b_k^* + \mu_{k\,k-1} b_{k-1}^*|^2 \geqq \tfrac{3}{4} |b_{k-1}^*|^2. \tag{1.20}$$

In this case we first achieve that

$$|\mu_{kj}| \leqq \tfrac{1}{2} \quad \text{for} \quad 1 \leqq j \leqq k-1. \tag{1.21}$$

[For $j=k-1$ this is already true, by (1.18).] If (1.21) does not hold, let l be the largest index $< k$ with $|\mu_{kl}| > \frac{1}{2}$, let r be the integer nearest to μ_{kl}, and replace b_k by

$b_k - rb_l$. The numbers μ_{kj} with $j<l$ are then replaced by $\mu_{kj}-r\mu_{lj}$, and μ_{kl} by $\mu_{kl}-r$; the other μ_{ij} and all b_i^* are unchanged. This is repeated until (1.21) holds.

Next we replace k by $k+1$. Then we are in the situation described by (1.16) and (1.17), and we proceed with the algorithm from there.

Notice that in the case $k=1$ we have done no more than replacing k by 2.

This finishes the description of the algorithm. Below we shall prove that the algorithm terminates.

(1.22) For the sake of completeness we now give the formulae that are needed in case 1. Let $b_1, b_2, \ldots, b_n$ be the current basis and b_i^*, μ_{ij} as in (1.2) and (1.3). Let k be the current subscript for which (1.16), (1.17), (1.18), and (1.19) hold. By c_i, c_i^*, and ν_{ij} we denote the vectors and numbers that will replace b_i, b_i^*, and μ_{ij}, respectively. The new basis $c_1, c_2, \ldots, c_n$ is given by

$$c_{k-1}=b_k, \quad c_k=b_{k-1}, \quad c_i=b_i \quad \text{for} \quad i\neq k-1, k.$$

Since c_{k-1}^* is the projection of b_k on the orthogonal complement of $\sum_{j=1}^{k-2} \mathbb{R}b_j$ we have, as announced:

$$c_{k-1}^*=b_k^*+\mu_{k\,k-1}b_{k-1}^*$$

[cf. the remark after (1.5)]. To obtain c_k^* we must project b_{k-1}^* on the orthogonal complement of $\mathbb{R}c_{k-1}^*$. That leads to

$$\begin{aligned}\nu_{k\,k-1}&=(b_{k-1}^*, c_{k-1}^*)/(c_{k-1}^*, c_{k-1}^*)\\&=\mu_{k\,k-1}|b_{k-1}^*|^2/|c_{k-1}^*|^2,\\c_k^*&=b_{k-1}^*-\nu_{k\,k-1}c_{k-1}^*.\end{aligned}$$

For $i\neq k-1, k$ we have $c_i^*=b_i^*$. Let now $i>k$. To find $\nu_{i\,k-1}$ and ν_{ik} we substitute

$$\begin{aligned}b_{k-1}^*&=\nu_{k\,k-1}c_{k-1}^*+c_k^*\\b_k^*&=(1-\mu_{k\,k-1}\nu_{k\,k-1})c_{k-1}^*-\mu_{k\,k-1}c_k^*\\&=(|b_k^*|^2/|c_{k-1}^*|^2)\cdot c_{k-1}^*-\mu_{k\,k-1}c_k^*\end{aligned}$$

in $b_i=b_i^*+\sum_{j=1}^{i-1}\mu_{ij}b_j^*$. That yields

$$\begin{aligned}\nu_{i\,k-1}&=\mu_{i\,k-1}\nu_{k\,k-1}+\mu_{ik}|b_k^*|^2/|c_{k-1}^*|^2\\\nu_{ik}&=\mu_{i\,k-1}-\mu_{ik}\mu_{k\,k-1}.\end{aligned}$$

Finally, we have

$$\nu_{k-1\,j}=\mu_{kj}, \quad \nu_{kj}=\mu_{k-1\,j}$$

for $1\leqq j<k-1$, and $\nu_{ij}=\mu_{ij}$ if $1\leqq j<i\leqq n$, $\{i,j\}\cap\{k-1, k\}=\emptyset$.

We remark that after the initialization stage of the algorithm it is not necessary to keep track of the vectors b_i^*. It suffices to keep track of the numbers $|b_i^*|^2$, in addition to μ_{ij} and the vectors b_i. Notice that $|c_k^*|^2=|b_{k-1}^*|^2\cdot|b_k^*|^2/|c_{k-1}^*|^2$ in the above, and that the left hand side of (1.19), (1.20) equals $|b_k^*|^2+\mu_{k\,k-1}^2|b_{k-1}^*|^2$.

The entire algorithm is represented in Fig. 1, in which $B_i=|b_i^*|^2$.

$b_i^* := b_i$;

$\left.\begin{array}{l}\mu_{ij} := (b_i, b_j^*)/B_j;\\ b_i^* := b_i^* - \mu_{ij} b_j^*\end{array}\right\}$ for $j = 1, 2, \ldots, i-1$;

$B_i := (b_i^*, b_i^*)$

$\Big\}$ for $i = 1, 2, \ldots, n$;

$k := 2$;

(1) perform (*) for $l = k-1$;

if $B_k < (\frac{3}{4} - \mu_{k\,k-1}^2) B_{k-1}$, go to (2);

perform (*) for $l = k-2, k-3, \ldots, 1$;

if $k = n$, terminate;

$k := k+1$;

go to (1);

(2) $\mu := \mu_{k\,k-1}$; $B := B_k + \mu^2 B_{k-1}$; $\mu_{k\,k-1} := \mu B_{k-1}/B$;

$B_k := B_{k-1} B_k / B$; $B_{k-1} := B$;

$\begin{pmatrix} b_{k-1} \\ b_k \end{pmatrix} := \begin{pmatrix} b_k \\ b_{k-1} \end{pmatrix}$;

$\begin{pmatrix} \mu_{k-1\,j} \\ \mu_{kj} \end{pmatrix} := \begin{pmatrix} \mu_{kj} \\ \mu_{k-1\,j} \end{pmatrix}$ for $j = 1, 2, \ldots, k-2$;

$\begin{pmatrix} \mu_{i\,k-1} \\ \mu_{ik} \end{pmatrix} := \begin{pmatrix} 1 & \mu_{k\,k-1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -\mu \end{pmatrix} \begin{pmatrix} \mu_{i\,k-1} \\ \mu_{ik} \end{pmatrix}$ for $i = k+1, k+2, \ldots, n$;

if $k > 2$, then $k := k-1$;

go to (1).

(*) If $|\mu_{kl}| > \frac{1}{2}$, then:

$\begin{cases} r := \text{integer nearest to } \mu_{kl};\ b_k := b_k - r b_l; \\ \mu_{kj} := \mu_{kj} - r\mu_{lj} \quad \text{for } j = 1, 2, \ldots, l-1; \\ \mu_{kl} := \mu_{kl} - r. \end{cases}$

Fig. 1. The reduction algorithm

(1.23) To prove that the algorithm terminates we introduce the quantities

$$d_i = \det((b_j, b_l))_{1 \leqq j, l \leqq i} \tag{1.24}$$

for $0 \leqq i \leqq n$. It is easily checked that

$$d_i = \prod_{j=1}^{i} |b_j^*|^2 \tag{1.25}$$

for $0 \leqq i \leqq n$. Hence the d_i are positive real numbers. Notice that $d_0 = 1$ and $d_n = d(L)^2$. Put

$$D = \prod_{i=1}^{n-1} d_i.$$

By (1.25), the number D only changes if some b_i^* is changed, which only occurs in case 1. In case 1, the number d_{k-1} is reduced by a factor $< \frac{3}{4}$, by (1.25), whereas the other d_i are unchanged, by (1.24); hence D is reduced by a factor $< \frac{3}{4}$. Below we prove that there is a positive lower bound for d_i that only depends on L. It follows

that there is also a positive lower bound for D, and hence an upper bound for the number of times that we pass through case 1.

In case 1, the value of k is decreased by 1, and in case 2 it is increased by 1. Initially we have $k=2$, and $k \leqq n+1$ throughout the algorithm. Therefore the number of times that we pass through case 2 is at most $n-1$ more than the number of times that we pass through case 1, and consequently it is bounded. This implies that the algorithm terminates.

To prove that d_i has a lower bound we put

$$m(L)=\min\{|x|^2 : x \in L, x \neq 0\}.$$

This is a positive real number. For $i>0$, we can interpret d_i as the square of the determinant of the lattice of rank i spanned by $b_1, b_2, \ldots, b_i$ in the vector space $\sum_{j=1}^{i} \mathbb{R}b_j$. By [4, Chap. I, Lemma 4 and Chap. II, Theorem I], this lattice contains a non-zero vector x with $|x|^2 \leqq (4/3)^{(i-1)/2} d_i^{1/i}$. Therefore $d_i \geqq (3/4)^{i(i-1)/2} m(L)^i$, as required.

We shall now analyse the running time of the algorithm under the added hypothesis that $b_i \in \mathbb{Z}^n$ for $1 \leqq i \leqq n$. By an *arithmetic operation* we mean an addition, subtraction, multiplication or division of two integers. Let the *binary length* of an integer a be the number of binary digits of $|a|$.

(1.26) **Proposition.** *Let $L \subset \mathbb{Z}^n$ be a lattice with basis $b_1, b_2, \ldots, b_n$, and let $B \in \mathbb{R}$, $B \geqq 2$, be such that $|b_i|^2 \leqq B$ for $1 \leqq i \leqq n$. Then the number of arithmetic operations needed by the basis reduction algorithm described in* (1.15) *is $O(n^4 \log B)$, and the integers on which these operations are performed each have binary length $O(n \log B)$.*

Remark. Using the classical algorithms for the arithmetic operations we find that the number of bit operations needed by the basis reduction algorithm is $O(n^6(\log B)^3)$. This can be reduced to $O(n^{5+\varepsilon}(\log B)^{2+\varepsilon})$, for every $\varepsilon>0$, if we employ fast multiplication techniques.

Proof of (1.26). We first estimate the number of times that we pass through cases 1 and 2. In the beginning of the algorithm we have $d_i \leqq B^i$, by (1.25), so $D \leqq B^{n(n-1)/2}$. Throughout the algorithm we have $D \geqq 1$, since $d_i \in \mathbb{Z}$ by (1.24) and $d_i>0$ by (1.25). So by the argument in (1.23) the number of times that we pass through case 1 is $O(n^2 \log B)$, and the same applies to case 2.

The initialization of the algorithm takes $O(n^3)$ arithmetic operations with rational numbers; below we shall see how they can be replaced by operations with integers.

For (1.18) we need $O(n)$ arithmetic operations, and this is also true for case 1. In case 2 we have to deal with $O(n)$ values of l, that each require $O(n)$ arithmetic operations. Since we pass through these cases $O(n^2 \log B)$ times we arrive at a total of $O(n^4 \log B)$ arithmetic operations.

In order to represent all numbers that appear in the course of the algorithm by means of *integers* we also keep track of the numbers d_i defined by (1.24). In the initialization stage these can be calculated by (1.25). After that, they are only changed in case 1. In that case, d_{k-1} is replaced by $d_{k-1} \cdot |c^*_{k-1}|^2 / |b^*_{k-1}|^2 = d_{k-2} \cdot |c^*_{k-1}|^2$ [in the notation of (1.22)] whereas the other d_i are unchanged. By (1.24),

the d_i are *integers*, and we shall now see that they can be used as denominators for all numbers that appear:

$$|b_i^*|^2 = d_i/d_{i-1} \quad (1 \leqq i \leqq n), \tag{1.27}$$

$$d_{i-1} b_i^* \in L \subset \mathbb{Z}^n \quad (1 \leqq i \leqq n), \tag{1.28}$$

$$d_j \mu_{ij} \in \mathbb{Z} \quad (1 \leqq j < i \leqq n). \tag{1.29}$$

The first of these follows from (1.25). For the second, we write $b_i^* = b_i - \sum_{j=1}^{i-1} \lambda_{ij} b_j$ with $\lambda_{ij} \in \mathbb{R}$. Solving $\lambda_{i1}, \ldots, \lambda_{i\,i-1}$ from the system

$$(b_i, b_l) = \sum_{j=1}^{i-1} \lambda_{ij} (b_j, b_l) \quad (1 \leqq l \leqq i-1)$$

and using (1.24) we find that $d_{i-1}\lambda_{ij} \in \mathbb{Z}$, whence (1.28). Notice that the same argument yields

$$d_{i-1}\left(b_k - \sum_{j=1}^{i-1} \mu_{kj} b_j^*\right) \in \mathbb{Z}^n \quad \text{for} \quad i \leqq k;$$

this is useful for the calculation of b_k^* at the beginning of the algorithm. To prove (1.29) we use (1.3), (1.27), and (1.28):

$$d_j \mu_{ij} = d_j (b_i, b_j^*)/(b_j^*, b_j^*) = d_{j-1}(b_i, b_j^*) = (b_i, d_{j-1} b_j^*) \in \mathbb{Z}.$$

To finish the proof of (1.26) we estimate all integers that appear. Since no d_i is ever increased we have $d_i \leqq B^i$ throughout the algorithm. This estimates the denominators. To estimate the numerators it suffices to find upper bounds for $|b_i^*|^2$, $|b_i|^2$, and $|\mu_{ij}|$.

At the beginning we have $|b_i^*|^2 \leqq |b_i|^2 \leqq B$, and $\max\{|b_i^*|^2 : 1 \leqq i \leqq n\}$ is non-increasing; to see this, use that $|c_{k-1}^*|^2 < \frac{3}{4}|b_{k-1}^*|^2$ and $|c_k^*|^2 \leqq |b_{k-1}^*|^2$ in (1.22), the latter inequality because c_k^* is a projection of b_{k-1}^*. Hence we have $|b_i^*|^2 \leqq B$ throughout the algorithm.

To deal with $|b_i|^2$ and μ_{ij} we first prove that every time we arrive at the situation described by (1.16) and (1.17) the following inequalities are satisfied:

$$|b_i|^2 \leqq nB \quad \text{for} \quad i \neq k, \tag{1.30}$$

$$|b_k|^2 \leqq n^2 (4B)^n \quad \text{if} \quad k \neq n+1, \tag{1.31}$$

$$|\mu_{ij}| \leqq \tfrac{1}{2} \quad \text{for} \quad 1 \leqq j < i, \; i < k, \tag{1.32}$$

$$|\mu_{ij}| \leqq (nB^j)^{1/2} \quad \text{for} \quad 1 \leqq j < i, \; i > k, \tag{1.33}$$

$$|\mu_{kj}| \leqq 2^{n-k}(nB^{n-1})^{1/2} \quad \text{for} \quad 1 \leqq j < k, \quad \text{if} \quad k \neq n+1. \tag{1.34}$$

Here (1.30), for $i < k$, is trivial from (1.32), and (1.31) follows from (1.34). Using that

$$\mu_{ij}^2 \leqq |b_i|^2/|b_j^*|^2 = d_{j-1}|b_i|^2/d_j \leqq B^{j-1}|b_i|^2 \tag{1.35}$$

we see that (1.33) follows from (1.30), and (1.32) is the same as (1.16). It remains to prove (1.30) for $i > k$ and to prove (1.34). At the beginning of the algorithm we even have $|b_i|^2 \leqq B$ and $\mu_{ij}^2 \leqq B^j$, by (1.35), so it suffices to consider the situation at the

end of cases 1 and 2. Taking into account that k changes in these cases, we see that in case 1 the set of vectors $\{b_i : i \neq k\}$ is unchanged, and that in case 2 the set $\{b_i : i > k\}$ is replaced by a subset. Hence the inequalities (1.30) are preserved. At the end of case 2, the new values for μ_{kj} (if $k \neq n+1$) are the old values of $\mu_{k+1\,j}$, so here (1.34) follows from the inequality (1.33) at the previous stage. To prove (1.34) at the end of case 1 we assume that it is valid at the previous stage, and we follow what happens to μ_{kj}. To achieve (1.18) it is, for $j < k-1$, replaced by $\mu_{kj} - r\mu_{k-1\,j}$, with $|r| < 2|\mu_{k\,k-1}|$ and $|\mu_{k-1\,j}| \leqq \frac{1}{2}$, so

$$|\mu_{kj} - r\mu_{k-1\,j}| \leqq |\mu_{kj}| + |\mu_{k\,k-1}| \leqq 2^{n-k+1}(nB^{n-1})^{1/2} \quad \text{by (1.34)}. \tag{1.36}$$

In the notation of (1.22) we therefore have

$$|v_{k-1\,j}| \leqq 2^{n-(k-1)}(nB^{n-1})^{1/2} \quad \text{for} \quad j < k-1$$

and since $k-1$ is the new value for k this is exactly the inequality (1.34) to be proved.

Finally, we have to estimate $|b_i|^2$ and μ_{ij} at the other points in the algorithm. For this it suffices to remark that the maximum of $|\mu_{k1}|, |\mu_{k2}|, \ldots, |\mu_{k\,k-1}|$ is at most doubled when (1.18) is achieved, by (1.36), and that the same thing happens in case 2 for at most $k-2$ values of l. Combining this with (1.34) and (1.33) we conclude that throughout the course of the algorithm we have

$$|\mu_{ij}| \leqq 2^{n-1}(nB^{n-1})^{1/2} \quad \text{for} \quad 1 \leqq j < i \leqq n$$

and therefore

$$|b_i|^2 \leqq n^2(4B)^n \quad \text{for} \quad 1 \leqq i \leqq n.$$

This finishes the proof of (1.26).

(1.37) *Remark.* Let $1 \leqq n' \leqq n$. If k, in the situation described by (1.16) and (1.17), is for the first time equal to $n'+1$, then the first n' vectors $b_1, b_2, \ldots, b_{n'}$ form a reduced basis for the lattice of rank n' spanned by the first n' vectors of the initially given basis. This will be useful in Sect. 3.

(1.38) *Remark.* It is easily verified that, apart from some minor changes, the analysis of our algorithm remains valid if the condition $L \subset \mathbb{Z}^n$ is replaced by the condition that $(x, y) \in \mathbb{Z}$ for all $x, y \in L$; or, equivalently, that $(b_i, b_j) \in \mathbb{Z}$ for $1 \leqq i, j \leqq n$. The weaker condition that $(b_i, b_j) \in \mathbb{Q}$, for $1 \leqq i, j \leqq n$, is also sufficient, but in this case we should clear denominators before applying (1.26).

We close this section with two applications of our reduction algorithm. The first is to simultaneous diophantine approximation. Let n be a positive integer, $\alpha_1, \alpha_2, \ldots, \alpha_n$ real numbers, and $\varepsilon \in \mathbb{R}$, $0 < \varepsilon < 1$. It is a classical theorem [4, Sect.V.10] that there exist integers $p_1, p_2, \ldots, p_n, q$ satisfying

$$|p_i - q\alpha_i| \leqq \varepsilon \quad \text{for} \quad 1 \leqq i \leqq n,$$
$$1 \leqq q \leqq \varepsilon^{-n}.$$

We show that there exists a polynomial-time algorithm to find integers that satisfy a slightly weaker condition.

(1.39) **Proposition.** *There exists a polynomial-time algorithm that, given a positive integer n and rational numbers $\alpha_1, \alpha_2, \ldots, \alpha_n, \varepsilon$ satisfying $0<\varepsilon<1$, finds integers $p_1, p_2, \ldots, p_n, q$ for which*

$$|p_i - q\alpha_i| \leqq \varepsilon \quad \text{for} \quad 1 \leqq i \leqq n,$$
$$1 \leqq q \leqq 2^{n(n+1)/4}\varepsilon^{-n}.$$

Proof. Let L be the lattice of rank $n+1$ spanned by the columns of the $(n+1)\times(n+1)$-matrix

$$\begin{pmatrix} 1 & 0 & \ldots & 0 & -\alpha_1 \\ 0 & 1 & \ldots & 0 & -\alpha_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & -\alpha_n \\ 0 & 0 & \ldots & 0 & 2^{-n(n+1)/4}\varepsilon^{n+1} \end{pmatrix}.$$

The inner product of any two columns is rational, so by (1.38) there is a polynomial-time algorithm to find a reduced basis $b_1, b_2, \ldots, b_{n+1}$ for L. By (1.9) we then have

$$|b_1| \leqq 2^{n/4} \cdot d(L)^{1/(n+1)} = \varepsilon.$$

Since $b_1 \in L$, we can write

$$b_1 = (p_1 - q\alpha_1, p_2 - q\alpha_2, \ldots, p_n - q\alpha_n, q \cdot 2^{-n(n+1)/4}\varepsilon^{n+1})^{\top}$$

with $p_1, p_2, \ldots, p_n, q \in \mathbb{Z}$. It follows that

$$|p_i - q\alpha_i| \leqq \varepsilon \quad \text{for} \quad 1 \leqq i \leqq n,$$
$$|q| \leqq 2^{n(n+1)/4}\varepsilon^{-n}.$$

From $\varepsilon<1$ and $b_1 \neq 0$ we see that $q \neq 0$. Replacing b_1 by $-b_1$, if necessary, we can achieve that $q>0$.

This proves (1.39).

Another application of our reduction algorithm is to the problem of finding $\mathbb{Q}$-linear relations among given real numbers $\alpha_1, \alpha_2, \ldots, \alpha_n$. For this we take the lattice L to be $\mathbb{Z}^n$, embedded in $\mathbb{R}^{n+1}$ by

$$(m_1, m_2, \ldots, m_n) \mapsto \left(m_1, m_2, \ldots, m_n, c \sum_{i=1}^{n} m_i \alpha_i'\right);$$

here c is a large constant and α_i' is a good rational approximation to α_i. The first basis vector of a reduced basis of L will give rise to integers $m_1, m_2, \ldots, m_n$ that are not too large such that $\sum_{i=1}^{n} m_i \alpha_i$ is very small.

Applying this to $\alpha_i = \alpha^{i-1}$ we see that our algorithm can be used to test a given real number α for algebraicity, and to determine its irreducible polynomial. Taking for α a zero of a polynomial $f \in \mathbb{Z}[X]$, $f \neq 0$, and generalizing the algorithm to *complex* α, one finds in this way an irreducible factor of f in $\mathbb{Z}[X]$. It is likely that this yields actually a polynomial-time algorithm to factor f in $\mathbb{Q}[X]$, an algorithm that is different from the p-adic method described in Sect. 3.

In a similar way we can test given real numbers $\alpha, \beta, \gamma, \ldots$ for algebraic dependence, taking the α_i to be the monomials in $\alpha, \beta, \gamma, \ldots$ up to a given degree.

2. Factors and Lattices

In this section we denote by p a prime number and by k a positive integer. We write $\mathbb{Z}/p^k\mathbb{Z}$ for the ring of integers modulo p^k, and $\mathbb{F}_p$ for the field $\mathbb{Z}/p\mathbb{Z}$. For $g=\sum_i a_i X^i \in \mathbb{Z}[X]$ we denote by $(g \bmod p^k)$ the polynomial $\sum_i (a_i \bmod p^k) X^i \in (\mathbb{Z}/p^k\mathbb{Z})[X]$.

We fix a polynomial $f \in \mathbb{Z}[X]$ of degree n, with $n>0$, and a polynomial $h \in \mathbb{Z}[X]$ that has the following properties:

(2.1) h has leading coefficient 1,

(2.2) $(h \bmod p^k)$ divides $(f \bmod p^k)$ in $(\mathbb{Z}/p^k\mathbb{Z})[X]$,

(2.3) $(h \bmod p)$ is irreducible in $\mathbb{F}_p[X]$,

(2.4) $(h \bmod p)^2$ does not divide $(f \bmod p)$ in $\mathbb{F}_p[X]$.

We put $l=\deg(h)$; so $0<l \leqq n$.

(2.5) **Proposition.** *The polynomial f has an irreducible factor h_0 in $\mathbb{Z}[X]$ for which $(h \bmod p)$ divides $(h_0 \bmod p)$, and this factor is uniquely determined up to sign. Further, if g divides f in $\mathbb{Z}[X]$, then the following three assertions are equivalent:*

(i) *$(h \bmod p)$ divides $(g \bmod p)$ in $\mathbb{F}_p[X]$,*

(ii) *$(h \bmod p^k)$ divides $(g \bmod p^k)$ in $(\mathbb{Z}/p^k\mathbb{Z})[X]$,*

(iii) *h_0 divides g in $\mathbb{Z}[X]$.*

In particular $(h \bmod p^k)$ divides $(h_0 \bmod p^k)$ in $(\mathbb{Z}/p^k\mathbb{Z})[X]$.

Proof. The existence of h_0 follows from (2.2) and (2.3), and the uniqueness, up to ± 1, from (2.4). The implications (ii) $\Rightarrow$ (i) and (iii) $\Rightarrow$ (i) are obvious. Now assume (i); we prove (iii) and (ii). From (i) and (2.4) it follows that $(h \bmod p)$ does not divide $(f/g \bmod p)$ in $\mathbb{F}_p[X]$. Therefore h_0 does not divide f/g in $\mathbb{Z}[X]$, so it must divide g. This proves (iii). By (2.3) the polynomials $(h \bmod p)$ and $(f/g \bmod p)$ are relatively prime in $\mathbb{F}_p[X]$, so in $\mathbb{F}_p[X]$ we have

$$(\lambda_1 \bmod p)\cdot(h \bmod p)+(\mu_1 \bmod p)\cdot(f/g \bmod p)=1$$

for certain $\lambda_1, \mu_1 \in \mathbb{Z}[X]$. Therefore $\lambda_1 h+\mu_1 f/g=1-pv_1$ for some $v_1 \in \mathbb{Z}[X]$. Multiplying this by $1+pv_1+p^2v_1^2+\ldots+p^{k-1}v_1^{k-1}$ and by g we obtain

$$\lambda_2 h+\mu_2 f \equiv g \bmod p^k\mathbb{Z}[X]$$

for certain $\lambda_2, \mu_2 \in \mathbb{Z}[X]$. Since the left hand side, when taken modulo p^k, is divisible by $(h \bmod p^k)$, the same is true for the right hand side. This proves (ii).

The final assertion of (2.5) follows if we take $g=h_0$. This proves (2.5).

(2.6) In the remainder of this section we fix an integer m with $m \geqq l$, and we let L be the collection of all polynomials in $\mathbb{Z}[X]$ of degree $\leqq m$ that, when taken modulo p^k, are divisible by $(h \bmod p^k)$ in $(\mathbb{Z}/p^k\mathbb{Z})[X]$. This is a subset of the $(m+1)$-dimensional real vector space $\mathbb{R}+\mathbb{R}\cdot X+\ldots+\mathbb{R}\cdot X^m$. This vector space is identified with $\mathbb{R}^{m+1}$ by identifying $\sum_{i=0}^{m} a_i X^i$ with $(a_0, a_1, \ldots, a_m)$. Notice that the length $\left|\sum_{i=0}^{m} a_i X^i\right|$ of a

polynomial, as defined in the introduction, is equal to the ordinary Euclidean length of $(a_0, a_1, \ldots, a_m)$. It is easy to see that L is a lattice in $\mathbb{R}^{m+1}$ and, using (2.1), that a basis of L is given by

$$\{p^k X^i : 0 \leqq i < l\} \cup \{hX^j : 0 \leqq j \leqq m-l\}.$$

From (1.1) it follows that $d(L) = p^{kl}$.

In the following proposition h_0 is as in (2.5).

(2.7) **Proposition.** *Let $b \in L$ satisfy*

$$p^{kl} > |f|^m \cdot |b|^n. \tag{2.8}$$

Then b is divisible by h_0 in $\mathbb{Z}[X]$, and in particular $\gcd(f, b) \neq 1$.

Remark. A weaker version of (2.7), which could also be used to obtain a polynomial-time factoring algorithm for polynomials, asserts that $\gcd(f, b) \neq 1$ under the same conditions. The proof of this version is less complicated than the proof given below, see [8, Theorem 2].

Proof of (2.7). We may assume that $b \neq 0$. Let $g = \gcd(f, b)$. By (2.5) it suffices to show that $(h \bmod p)$ divides $(g \bmod p)$. Suppose that this is not the case. Then by (2.3) we have

$$\lambda_3 h + \mu_3 g = 1 - p\nu_3 \tag{2.9}$$

for certain $\lambda_3, \mu_3, \nu_3 \in \mathbb{Z}[X]$. We shall derive a contradiction from this.

Put $e = \deg(g)$ and $m' = \deg(b)$. Clearly $0 \leqq e \leqq m' \leqq m$. We define

$$\begin{aligned} M &= \{\lambda f + \mu b : \lambda, \mu \in \mathbb{Z}[X], \deg(\lambda) < m' - e, \deg(\mu) < n - e\} \\ &\subset \mathbb{Z} + \mathbb{Z} \cdot X + \ldots + \mathbb{Z} \cdot X^{n+m'-e-1}. \end{aligned}$$

Let M' be the projection of M on

$$\mathbb{Z} \cdot X^e + \mathbb{Z} \cdot X^{e+1} + \ldots + \mathbb{Z} \cdot X^{n+m'-e-1}.$$

Suppose that $\lambda f + \mu b$ projects to 0 in M', with λ, μ as in the definition of M. Then $\deg(\lambda f + \mu b) < e$, but g divides $\lambda f + \mu b$, so $\lambda f + \mu b = 0$. From $\lambda \cdot (f/g) = -\mu \cdot (b/g)$ and $\gcd(f/g, b/g) = 1$ it follows that f/g divides μ. But $\deg(\mu) < n - e = \deg(f/g)$, so $\mu = 0$, and therefore also $\lambda = 0$.

This proves that the projections of

$$\{X^i f : 0 \leqq i < m' - e\} \cup \{X^j b : 0 \leqq j < n - e\}$$

on M' are linearly independent. Since these projections span M', it follows that M' is a lattice of rank $n + m' - 2e$. From Hadamard's inequality (1.10) and (2.8) we obtain

$$d(M') \leqq |f|^{m'-e} \cdot |b|^{n-e} \leqq |f|^m \cdot |b|^n < p^{kl}. \tag{2.10}$$

Below we deduce from (2.9) that

$$\{v \in M : \deg(v) < e + l\} \subset p^k \mathbb{Z}[X]. \tag{2.11}$$

Hence, if we choose a basis $b_e, b_{e+1}, \ldots, b_{n+m'-e-1}$ of M' with $\deg(b_j)=j$, see [4, Chap. I, Theorem I.A], then the leading coefficients of $b_e, b_{e+1}, \ldots, b_{e+l-1}$ are divisible by p^k. [Notice that $e+l-1 \leqq n+m'-e-1$ because g divides b and $(h \bmod p)$ divides $(f/g \bmod p)$.] Since $d(M')$ equals the absolute value of the product of the leading coefficients of $b_e, b_{e+1}, \ldots, b_{n+m'-e-1}$ we find that $d(M') \geqq p^{kl}$. Combined with (2.10) this is the desired contradiction.

To prove (2.11), let $v \in M$, $\deg(v) < e+l$. Then g divides v. Multiplying (2.9) by v/g and by $1+pv_3+p^2v_3^2+\ldots+p^{k-1}v_3^{k-1}$ we obtain

$$\lambda_4 h + \mu_4 v \equiv v/g \bmod p^k \mathbb{Z}[X] \tag{2.12}$$

with $\lambda_4, \mu_4 \in \mathbb{Z}[X]$. From $v \in M$ and $b \in L$ it follows that $(v \bmod p^k)$ is divisible by $(h \bmod p^k)$. So by (2.12) also $(v/g \bmod p^k)$ is divisible by $(h \bmod p^k)$. But $(h \bmod p^k)$ is of degree l with leading coefficient 1, while $(v/g \bmod p^k)$ has degree $< e+l-e=l$. Therefore $v/g \equiv 0 \bmod p^k \mathbb{Z}[X]$, so also $v \equiv 0 \bmod p^k \mathbb{Z}[X]$. This proves (2.11).

This concludes the proof of (2.7).

(2.13) **Proposition.** *Let p, k, f, n, h, l be as at the beginning of this section, h_0 as in (2.5), and m, L as in (2.6). Suppose that $b_1, b_2, \ldots, b_{m+1}$ is a reduced basis for L (see (1.4) and (1.5)), and that*

$$p^{kl} > 2^{mn/2} \binom{2m}{m}^{n/2} |f|^{m+n}. \tag{2.14}$$

Then we have $\deg(h_0) \leqq m$ if and only if

$$|b_1| < (p^{kl}/|f|^m)^{1/n}. \tag{2.15}$$

Proof. The "if"-part is immediate from (2.7), since $\deg(b_1) \leqq m$. To prove the "only if"-part, assume that $\deg(h_0) \leqq m$. Then $h_0 \in L$ by (2.5), and $|h_0| \leqq \binom{2m}{m}^{1/2} \cdot |f|$ by a result of Mignotte [10; cf. 7, Exercise 4.6.2.20]. Applying (1.11) to $x = h_0$ we find that $|b_1| \leqq 2^{m/2} \cdot |h_0| \leqq 2^{m/2} \cdot \binom{2m}{m}^{1/2} \cdot |f|$. By (2.14) this implies (2.15). This proves (2.13).

(2.16) **Proposition.** *Let the notation and the hypotheses be the same as in (2.13), and assume in addition that there exists an index $j \in \{1, 2, \ldots, m+1\}$ for which*

$$|b_j| < (p^{kl}/|f|^m)^{1/n}. \tag{2.17}$$

Let t be the largest such j. Then we have

$$\deg(h_0) = m+1-t,$$

$$h_0 = \gcd(b_1, b_2, \ldots, b_t),$$

and (2.17) holds for all j with $1 \leqq j \leqq t$.

Proof. Let $J = \{j \in \{1, 2, \ldots, m+1\}$: (2.17) holds$\}$. From (2.7) we know that h_0 divides b_j for every $j \in J$. Hence if we put

$$h_1 = \gcd(\{b_j : j \in J\})$$

then h_0 divides h_1. Each $b_j, j \in J$, is divisible by h_1 and has degree $\leqq m$, so belongs to

$$\mathbb{Z} \cdot h_1 + \mathbb{Z} \cdot h_1 X + \ldots + \mathbb{Z} \cdot h_1 X^{m - \deg(h_1)}.$$

Since the b_j are linearly independent this implies that

$$\# J \leqq m + 1 - \deg(h_1). \tag{2.18}$$

By the result of Mignotte used in the proof of (2.13) we have $|h_0 X^i| = |h_0| \leqq \binom{2m}{m}^{1/2} \cdot |f|$ for all $i \geqq 0$. For $i = 0, 1, \ldots, m - \deg(h_0)$ we have $h_0 X^i \in L$, so from (1.12) we obtain

$$|b_j| \leqq 2^{m/2} \cdot \binom{2m}{m}^{1/2} \cdot |f|$$

for $1 \leqq j \leqq m + 1 - \deg(h_0)$. By (2.14), this implies that

$$\{1, 2, \ldots, m + 1 - \deg(h_0)\} \subset J. \tag{2.19}$$

From (2.18), (2.19) and the fact that h_0 divides h_1 we now see that equality must hold in (2.18) and (2.19), and that

$$\deg(h_0) = \deg(h_1) = m + 1 - t, \qquad J = \{1, 2, \ldots, t\}.$$

It remains to prove that h_0 is equal to h_1, up to sign, and for this it suffices to check that h_1 is primitive. Choose $j \in J$, and let d_j be the content of b_j. Then b_j/d_j is divisible by h_0, and $h_0 \in L$, so $b_j/d_j \in L$. But b_j belongs to a basis for L, so $d_j = 1$ and b_j is primitive, and the same is true for the factor h_1 of b_j. This finishes the proof of (2.16).

Remark. If $t = 1$ then we see from (2.16) that b_1 is an irreducible factor of f, and that no gcd computation is necessary.

Remark. From the proofs of (2.13) and (2.16) we see that (2.14) may be replaced by

$$p^{kl} > \beta^n \gamma^n |f|^m,$$

where $\beta = \max\{|b_j|/|b_i^*| : 1 \leqq j \leqq i \leqq m + 1\}$ [cf. (1.14)] and where γ is such that $|g| \leqq \gamma$ for every factor g of f in $\mathbb{Z}[X]$ with $\deg(g) \leqq m$.

3. Description of the Algorithm

Denote by f a primitive polynomial in $\mathbb{Z}[X]$ of degree n, with $n > 0$. In this section we describe an algorithm that factors f into irreducible factors in $\mathbb{Z}[X]$. We begin with two auxiliary algorithms.

(3.1) Suppose that, in addition to f and n, a prime number p, a positive integer k and a polynomial $h \in \mathbb{Z}[X]$ are given satisfying (2.1), (2.2), (2.3), and (2.4). Assume that the coefficients of h are reduced modulo p^k, so

$$|h|^2 \leqq 1 + l p^{2k},$$

where $l=\deg(h)$. Let further an integer $m \geqq l$ be given, and assume that inequality (2.14) is satisfied:

$$p^{kl} > 2^{mn/2} \cdot \binom{2m}{m}^{n/2} \cdot |f|^{m+n}.$$

We describe an algorithm that decides whether $\deg(h_0) \leqq m$, with h_0 as in (2.5), and determines h_0 if indeed $\deg(h_0) \leqq m$.

Let L be the lattice defined in (2.6), with basis

$$\{p^k X^i : 0 \leqq i < l\} \cup \{hX^j : 0 \leqq j \leqq m-l\}.$$

Applying algorithm (1.15) we find a *reduced* basis $b_1, b_2, \ldots, b_{m+1}$ for L. If $|b_1| \geqq (p^{kl}/|f|^m)^{1/n}$ then by (2.13) we have $\deg(h_0) > m$, and the algorithm stops. If $|b_1| < (p^{kl}/|f|^m)^{1/n}$ then by (2.13) and (2.16) we have $\deg(h_0) \leqq m$ and

$$h_0 = \gcd(b_1, b_2, \ldots, b_t)$$

with t as in (2.16). This gcd can be calculated by repeated application of the subresultant algorithm described in [7, Sect. 4.6.1]. This finishes the description of algorithm (3.1).

(3.2) **Proposition.** *The number of arithmetic operations needed by algorithm* (3.1) *is* $O(m^4 k \log p)$, *and the integers on which these operations are performed each have binary length* $O(mk \log p)$.

Proof. We apply (1.26) with $m+1$ in the role of n and with $B = 1 + lp^{2k}$. From $l \leqq n$ and (2.14) we see that $m = O(k \log p)$, so $\log l < l \leqq m$ implies that $\log B = O(k \log p)$. This leads to the estimates in (3.2). It is straightforward to verify that the gcd computation at the end satisfies the same estimates. This proves (3.2).

(3.3) Next suppose that, in addition to f and n, a prime number p and a polynomial $h \in \mathbb{Z}[X]$ are given such that (2.1), (2.2), (2.3), and (2.4) are satisfied with k replaced by 1. Assume that the coefficients of h are reduced modulo p. We describe an algorithm that determines h_0, the irreducible factor of f for which $(h \bmod p)$ divides $(h_0 \bmod p)$, cf. (2.5).

Write $l = \deg(h)$. If $l = n$ then $h_0 = f$, and the algorithm stops. Let now $l < n$. We first calculate the least positive integer k for which (2.14) holds with m replaced by $n-1$:

$$p^{kl} > 2^{(n-1)n/2} \cdot \binom{2(n-1)}{n-1}^{n/2} \cdot |f|^{2n-1}.$$

Next we modify h, without changing $(h \bmod p)$, in such a way that (2.2) holds for the value of k just calculated, in addition to (2.1), (2.3), and (2.4). This can be accomplished by the use of Hensel's lemma, see [7, Exercise 4.6.2.22; 14; 15; 13]. We may assume that the coefficients of h are reduced modulo p^k.

Let u be the greatest integer for which $l \leqq (n-1)/2^u$. We perform algorithm (3.1) for each of the values $m = [(n-1)/2^u]$, $[(n-1)/2^{u-1}]$, $\ldots, [(n-1)/2]$, $n-1$ in succession, with $[x]$ denoting the greatest integer $\leqq x$; but we stop as soon as for one of these values of m algorithm (3.1) succeeds in determining h_0. If this does not occur for any m in the sequence then $\deg(h_0) > n-1$, so $h_0 = f$ and we stop. This finishes the description of algorithm (3.3).

(3.4) **Proposition.** *Denote by* $m_0 = \deg(h_0)$ *the degree of the irreducible factor* h_0 *of* f *that is found by algorithm* (3.3). *Then the number of arithmetic operations needed by algorithm* (3.3) *is* $O(m_0(n^5 + n^4 \log|f| + n^3 \log p))$, *and the integers on which these operations are performed each have binary length* $O(n^3 + n^2 \log|f| + n \log p)$.

Proof. From

$$p^{k-1} \leqq p^{(k-1)l} \leqq 2^{(n-1)n/2} \binom{2(n-1)}{n-1}^{n/2} |f|^{2n-1}$$

it follows that

$$k \log p = (k-1) \log p + \log p = O(n^2 + n \log|f| + \log p).$$

Let m_1 be the largest value of m for which algorithm (3.1) is performed. From the choice of values for m it follows that $m_1 < 2m_0$, and that every other value for m that is tried is of the form $[m_1/2^i]$, with $i \geqq 1$. Therefore we have $\sum m^4 = O(m_0^4)$. Using (3.2) we conclude that the total number of arithmetic operations needed by the applications of algorithm (3.1) is $O(m_0^4 k \log p)$, which is

$$O(m_0^4(n^2 + n \log|f| + \log p)),$$

and that the integers involved each have binary length $O(m_1 k \log p)$, which is

$$O(m_0(n^2 + n \log|f| + \log p)).$$

With some care it can be shown that the same estimates are valid for a suitable version of Hensel's lemma. But it is simpler, and sufficient for our purpose, to replace the above estimates by the estimates stated in (3.4), using that $m_0 \leqq n$; then a very crude estimate for Hensel's lemma will do. The straightforward verification is left to the reader. This proves (3.4).

(3.5) We now describe an algorithm that factors a given primitive polynomial $f \in \mathbb{Z}[X]$ of degree $n > 0$ into irreducible factors in $\mathbb{Z}[X]$.

The first step is to calculate the resultant $R(f, f')$ of f and its derivative f', using the subresultant algorithm [7, Sect. 4.6.1]. If $R(f, f') = 0$ then f and f' have a greatest common divisor g in $\mathbb{Z}[X]$ of positive degree, and g is also calculated by the subresultant algorithm. This case will be discussed at the end of the algorithm. Assume now that $R(f, f') \neq 0$.

In the second step we determine the smallest prime number p not dividing $R(f, f')$, and we decompose $(f \bmod p)$ into irreducible factors in $\mathbb{F}_p[X]$ by means of Berlekamp's algorithm [7, Sect. 4.6.2]. Notice that $R(f, f')$ is, up to sign, equal to the product of the leading coefficient of f and the discriminant of f. So $R(f, f') \not\equiv 0 \bmod p$ implies that $(f \bmod p)$ still has degree n, and that it has no multiple factors in $\mathbb{F}_p[X]$. Therefore (2.4) is valid for every irreducible factor $(h \bmod p)$ of $(f \bmod p)$ in $\mathbb{F}_p[X]$.

In the third step we assume that we know a decomposition $f = f_1 f_2$ in $\mathbb{Z}[X]$ such that the complete factorizations of f_1 in $\mathbb{Z}[X]$ and $(f_2 \bmod p)$ in $\mathbb{F}_p[X]$ are known. At the start we can take $f_1 = 1$, $f_2 = f$. In this situation we proceed as follows. If $f_2 = \pm 1$ then $f = \pm f_1$ is completely factored in $\mathbb{Z}[X]$, and the algorithm stops. Suppose now that f_2 has positive degree, and choose an irreducible factor

$(h \bmod p)$ of $(f_2 \bmod p)$ in $\mathbb{F}_p[X]$. We may assume that the coefficients of h are reduced modulo p and that h has leading coefficient 1. Then we are in the situation described at the start of algorithm (3.3), with f_2 in the role of f, and we use that algorithm to find the irreducible factor h_0 of f_2 in $\mathbb{Z}[X]$ for which $(h \bmod p)$ divides $(h_0 \bmod p)$. We now replace f_1 and f_2 by $f_1 h_0$ and f_2/h_0, respectively, and from the list of irreducible factors of $(f_2 \bmod p)$ we delete those that divide $(h_0 \bmod p)$. After this we return to the beginning of the third step.

This finishes the description of the algorithm in the case that $R(f, f') \neq 0$. Suppose now that $R(f, f') = 0$, let g be the gcd of f and f' in $\mathbb{Z}[X]$, and put $f_0 = f/g$. Then f_0 has no multiple factors in $\mathbb{Z}[X]$, so $R(f_0, f_0') \neq 0$, and we can factor f_0 using the main part of the algorithm. Since each irreducible factor of g in $\mathbb{Z}[X]$ divides f_0 we can now complete the factorization of $f = f_0 g$ by a few trial divisions. This finishes the description of algorithm (3.5).

(3.6) **Theorem.** *The above algorithm factors any primitive polynomial $f \in \mathbb{Z}[X]$ of positive degree n into irreducible factors in $\mathbb{Z}[X]$. The number of arithmetic operations needed by the algorithm is $O(n^6 + n^5 \log|f|)$, and the integers on which these operations are performed each have binary length $O(n^3 + n^2 \log|f|)$. Here $|f|$ is as defined in the introduction.*

Using the classical algorithms for the arithmetic operations we now arrive at the bound $O(n^{12} + n^9(\log|f|)^3)$ for the number of bit operations that was announced in the introduction. This can be reduced to $O(n^{9+\varepsilon} + n^{7+\varepsilon}(\log|f|)^{2+\varepsilon})$, for every $\varepsilon > 0$, if we employ fast multiplication techniques.

Proof of (3.6). The correctness of the algorithm is clear from its description. To prove the estimates we first assume that $R(f, f') \neq 0$. We begin by deriving an upper bound for p. Since p is the least prime not dividing $R(f, f')$ we have

$$\prod_{q < p,\, q \text{ prime}} q \leqq |R(f, f')|. \tag{3.7}$$

It is not difficult to prove that there is a positive constant A such that

$$\prod_{q < p,\, q \text{ prime}} q > e^{Ap} \tag{3.8}$$

for all $p > 2$, see [6, Sect. 22.2]; by [12] we can take $A = 0.84$ for $p > 101$. From Hadamard's inequality (1.10) we easily obtain

$$|R(f, f')| \leqq n^n |f|^{2n-1}.$$

Combining this with (3.7) and (3.8) we conclude that

$$p < (n \log n + (2n-1) \log|f|)/A \tag{3.9}$$

or $p = 2$. Therefore the terms involving $\log p$ in proposition (3.4) are absorbed by the other terms.

The call of algorithm (3.3) in the third step requires $O(m_0 \cdot (n^5 + n^4 \log|f_2|))$ arithmetic operations, by (3.4), where m_0 is the degree of the factor h_0 that is found. Since f_2 divides f, Mignotte's theorem [10; cf. 7, Exercise 4.6.2.20] that was used in the proof of (2.13) implies that $\log|f_2| = O(n + \log|f|)$. Further the sum $\sum m_0$ of the

degrees of the irreducible factors of f is clearly equal to n. We conclude that the total number of arithmetic operations needed by the applications of (3.3) is $O(n^6 + n^5 \log|f|)$. By (3.4), the integers involved in (3.3) each have binary length $O(n^3 + n^2 \log|f|)$.

We must now show that the other parts of the algorithm satisfy the same estimates. For the subresultant algorithm in the first step and the remainder of the third step this is entirely straightforward and left to the reader. We consider the second step.

Write P for the right hand side of (3.9). Then p can be found with $O(P)$ arithmetic operations on integers of binary length $O(P)$; here one can apply [11] to generate a table of prime numbers $< P$, or alternatively use a table of squarefree numbers, which is easier to generate. From $p < P$ it also follows that Berlekamp's algorithm satisfies the estimates stated in the theorem, see [7, Sect. 4.6.2].

Finally, let $R(f, f') = 0$, and $f_0 = f/\gcd(f, f')$ as in the algorithm. Since f_0 divides f, Mignotte's theorem again implies that $\log|f_0| = O(n + \log|f|)$. The theorem now follows easily by applying the preceding case to f_0.

This finishes the proof of (3.6).

(3.10) For the algorithms described in this section the precise choice of the basis reduction algorithm is irrelevant, as long as it satisfies the estimates of proposition (1.26). A few simplifications are possible if the algorithm explained in Sect. 1 is used. Specifically, the gcd computation at the end of algorithm (3.1) can be avoided. To see this, assume that $m_0 = \deg(h_0)$ is indeed $\leqq m$. We claim that h_0 occurs as b_1 in the course of the basis reduction algorithm. Namely, by (1.37) it will happen at a certain moment that $b_1, b_2, \ldots, b_{m_0+1}$ form a reduced basis for the lattice of rank $m_0 + 1$ spanned by $\{p^k X^i : 0 \leqq i < l\} \cup \{hX^j : 0 \leqq j \leqq m_0 - l\}$. At that moment, we have $h_0 = b_1$, by (2.13) and (2.16), applied with m_0 in the role of m. A similar argument shows that in algorithm (3.3) one can simply try the values $m = l$, $l+1, \ldots, n-1$ in succession, until h_0 is found.

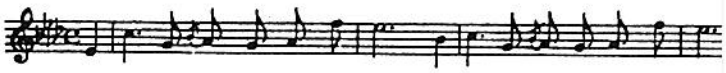
Acknowledgements are due to J. J. M. Cuppen for permission to include his improvement of our basis reduction algorithm in Sect. 1.

References

1. Adleman, L.M., Odlyzko, A.M.: Irreducibility testing and factorization of polynomials, to appear. Extended abstract: Proc. 22nd Annual IEEE Symp. Found. Comp. Sci., pp. 409–418 (1981)
2. Brentjes, A.J.: Multi-dimensional continued fraction algorithms. Mathematical Centre Tracts 145. Amsterdam: Mathematisch Centrum 1981
3. Cantor, D.G.: Irreducible polynomials with integral coefficients have succinct certificates. J. Algorithms **2**, 385–392 (1981)
4. Cassels, J.W.S.: An introduction to the geometry of numbers. Berlin, Heidelberg, New York: Springer 1971
5. Ferguson, H.R.P., Forcade, R.W.: Generalization of the Euclidean algorithm for real numbers to all dimensions higher than two. Bull. Am. Math. Soc. **1**, 912–914 (1979)
6. Hardy, G.H., Wright, E.M.: An introduction to the theory of numbers. Oxford: Oxford University Press 1979
7. Knuth, D.E.: The art of computer programming, Vol. 2, Seminumerical algorithms. Reading: Addison-Wesley 1981

8. Lenstra, A.K.: Lattices and factorization of polynomials, Report IW 190/81. Amsterdam: Mathematisch Centrum 1981
9. Lenstra, H.W., Jr.: Integer programming with a fixed number of variables. Math. Oper. Res. (to appear)
10. Mignotte, M.: An inequality about factors of polynomials. Math. Comp. **28**, 1153–1157 (1974)
11. Pritchard, P.: A sublinear additive sieve for finding prime numbers. Comm. ACM **24**, 18–23 (1981)
12. Barkley Rosser, J., Schoenfeld, L.: Approximate formulas for some functions of prime numbers. Ill. J. Math. **6**, 64–94 (1962)
13. Yun, D.Y.Y.: The Hensel lemma in algebraic manipulation. Cambridge: MIT 1974; reprint: New York: Garland 1980
14. Zassenhaus, H.: On Hensel factorization. I. J. Number Theory **1**, 291–311 (1969)
15. Zassenhaus, H.: A remark on the Hensel factorization method. Math. Comp. **32**, 287–292 (1978)
16. Zassenhaus, H.: A new polynomial factorization algorithm (unpublished manuscript, 1981)

Received July 11, 1982



Factoring polynomials over algebraic number fields

by

A.K. Lenstra

ABSTRACT

This paper describes a polynomial-time algorithm for the factorization of polynomials in one variable with coefficients in an algebraic number field. The algorithm generalizes the polynomial-time algorithm for the factorization of polynomials in one variable with rational coefficients.

KEY WORDS & PHRASES: *polynomial algorithm, polynomial factorization*

Factoring polynomials over algebraic number fields.

In [8] a polynomial-time algorithm was given to factorize polynomials in one variable with rational coefficients. In this paper we generalize this result to polynomials in one variable with coefficients in an algebraic number field.

The existence of a polynomial-time algorithm for this problem is not surprising in view of [8]. According to Trager [12] the problem is reducible to the factorization of univariate polynomials with integral coefficients, and in [6] it is shown that this reduction is polynomial-time. Here we pursue a direct approach to the factorization of polynomials over algebraic number fields. As suggested in [7: Section 5] we regard the irreducible factor we are looking for as an element of a certain integral lattice, and we prove that it is the 'smallest' element in this lattice. As we have seen in [8] this enables us to effectively compute this factor by means of a basis reduction algorithm for lattices.

Section 1 contains some notation and definitions; furthermore we recall there some results from [8: Section 1]. Section 2 deals with the connection between factors and lattices. It generalizes the first part of [8: Section 2]. In Section 3 we give a global description of the factoring algorithm and we analyze its running time.

For a polynomial f we denote by δf the *degree* of f, by $\ell c(f)$ the *leading coefficient* of f, and f is said to be *monic* if $\ell c(f) = 1$.

1. Preliminaries.

Let the algebraic number field $\mathbb{Q}(\alpha)$ be given as the field of rational numbers $\mathbb{Q}$ extended by a root α of a prescribed monic irreducible polynomial $F \in \mathbb{Z}[T]$, i.e. $\mathbb{Q}(\alpha) \simeq \mathbb{Q}[T]/(F)$. This implies that the elements of $\mathbb{Q}(\alpha)$ can be represented as polynomials in α over $\mathbb{Q}$ of degree $< \delta F$. We may assume that the degree of the *minimal polynomial* F is at least 2.

Similarly, we define $\mathbb{Z}[\alpha] = \mathbb{Z}[T]/(F)$ as the ring of polynomials in α over $\mathbb{Z}$ of degree $< \delta F$, where multiplication is done 'modulo F'.

Let f be a monic polynomial in $\mathbb{Q}(\alpha)[X]$. In Section 3 we will describe how to choose a positive integer D such that

(1.1) f and all monic factors of f in $\mathbb{Q}(\alpha)[X]$ are in $\frac{1}{D}\mathbb{Z}[\alpha][X]$.

The algorithm to determine the irreducible factors of f in $\mathbb{Q}(\alpha)[X]$ that we will present, is very similar to the algorithm for factorization in $\mathbb{Z}[X]$ as described in [8]: first determine the factorization of f over some finite field ($\mathbb{Z}/p\mathbb{Z}$ in [8]), next extend this factorization to a factorization over a large enough ring ($\mathbb{Z}/p^k\mathbb{Z}$ in [8]), and finally use a lattice reduction algorithm to determine the factors over $\mathbb{Q}(\alpha)$. Therefore, we first describe how to choose this finite field and this ring.

Let p be a prime number such that

(1.2) p does not divide D,

and let k be a positive integer. For $G = \Sigma_i a_i T^i \in \mathbb{Z}[T]$ and some integer ℓ we denote by G_ℓ or $(G \bmod p^\ell)$ the polynomial $\Sigma_i (a_i \bmod p^\ell) T^i \in (\mathbb{Z}/p^\ell\mathbb{Z})[T]$. In Section 3 we will see that we are able to determine p

in such a way that we can compute a polynomial $H \in \mathbb{Z}[T]$ such that

(1.3) H is monic,

(1.4) H_k divides F_k in $(\mathbb{Z}/p^k\mathbb{Z})[T]$,

(1.5) H_1 is irreducible in $(\mathbb{Z}/p\mathbb{Z})[T]$,

(1.6) $(H_1)^2$ does not divide F_1 in $(\mathbb{Z}/p\mathbb{Z})[T]$.

It follows that H_1 divides F_1 in $(\mathbb{Z}/p\mathbb{Z})[T]$, and that $0 < \delta H \leq \delta F$.

This polynomial H, together with the prime number p and the integer k, gives us the possibility to construct the finite field and the ring we were looking for. We denote by q the prime-power $p^{\delta H}$ and by $\mathbf{F}_q$ the finite field containing q elements. From (1.5) we derive that $\mathbf{F}_q \simeq (\mathbb{Z}/p\mathbb{Z})[T]/(H_1)$. Remark that $\mathbf{F}_q \simeq \{\Sigma_{i=0}^{\delta H-1} a_i\alpha_1^i : a_i \in \mathbb{Z}/p\mathbb{Z}\}$ where $\alpha_1 = (T \bmod (H_1))$ is a zero of H_1. This enables us to represent the elements of $\mathbf{F}_q$ as polynomials in α_1 over $\mathbb{Z}/p\mathbb{Z}$ of degree $< \delta H$. The finite field $\mathbf{F}_q$ corresponds to $\mathbb{Z}/p\mathbb{Z}$ in [8]; we now define the ring which will play the role of $\mathbb{Z}/p^k\mathbb{Z}$ in [8]. Let $W_k(\mathbf{F}_q) = (\mathbb{Z}/p^k\mathbb{Z})[T]/(H_k)$ be a ring containing q^k elements. We have that $W_k(\mathbf{F}_q) = \{\Sigma_{i=0}^{\delta H-1} a_i\alpha_k^i : a_i \in \mathbb{Z}/p^k\mathbb{Z}\}$ where $\alpha_k = (T \bmod (H_k))$ is a zero of H_k. So elements of $W_k(\mathbf{F}_q)$ can be represented as polynomials in α_k over $\mathbb{Z}/p^k\mathbb{Z}$ of degree $< \delta H$, and $W_k(\mathbf{F}_q)$ can be mapped onto $\mathbf{F}_q$ by reducing the coefficients of these polynomials modulo p. For $a \in W_k(\mathbf{F}_q)[X]$ we denote by $(a \bmod p) \in \mathbf{F}_q[X]$ the result of applying this mapping coefficient-wise to a. Remark that $W_1(\mathbf{F}_q) \simeq \mathbf{F}_q$.

We now show how we map polynomials in $\frac{1}{D}\mathbb{Z}[\alpha][X]$ to polynomials in $\mathbf{F}_q[X]$ and $W_k(\mathbf{F}_q)[X]$ respectively. Clearly, the canonical mapping from

$\mathbb{Z}[T]/(F)$ to $(\mathbb{Z}/p^\ell\mathbb{Z})[T]/(H_\ell)$ defines a mapping from $\mathbb{Z}[\alpha]$ to $W_\ell(\mathbb{F}_q)$, for $\ell = 1,k$. (Informally, this mapping works by reducing the polynomial in α modulo p^ℓ and $H_\ell(\alpha)$.) For $a \in \mathbb{Z}[\alpha]$ we denote by $(a \bmod (p^\ell, H_\ell)) \in W_\ell(\mathbb{F}_q)$ the result of this mapping. Finally, for $g = \Sigma_i \frac{a_i}{D} X^i \in \frac{1}{D}\mathbb{Z}[\alpha][X]$ we denote by $(g \bmod (p^\ell, H_\ell))$ the polynomial $\Sigma_i (((D^{-1} \bmod p^\ell) a_i) \bmod (p^\ell, H_\ell)) X^i \in W_\ell(\mathbb{F}_q)[X]$. Notice that $D^{-1} \bmod p^\ell$ exists due to (1.2).

(1.7) We conclude this section with some results from [8: Section 1]. Let n be a positive integer, and let $b_1, b_2, \ldots, b_n \in \mathbb{R}^n$ be linearly independent. The *lattice* $L \subset \mathbb{R}^n$ of *rank* n spanned by $b_1, b_2, \ldots, b_n$ is defined as

$$L = \Sigma_{i=1}^n \mathbb{Z}\, b_i = \{\Sigma_{i=1}^n r_i b_i : r_i \in \mathbb{Z} \ (1 \le i \le n)\}.$$

We assume that the $n \times n$ matrix having $b_1, b_2, \ldots, b_n$ as columns is upper-triangular, i.e. the $(j+1)$-th up to the n-th coordinate of b_j is zero, for $1 \le j \le n$. This implies that we can regard the lattice L_j of rank j spanned by $b_1, b_2, \ldots, b_j$ as a lattice contained in $\mathbb{R}^j$, for $1 \le j \le n$; notice that $L = L_n$. Furthermore we assume that $b_1, b_2, \ldots, b_n \in (\frac{1}{D}\mathbb{Z})^n$, so that $L_j \subset (\frac{1}{D}\mathbb{Z})^j$.

Let $B \in \mathbb{Z}_{\ge 2}$ be chosen in such a way that $|Db_i|^2 \le B$ for $1 \le i \le n$, where $|\ |$ denotes the ordinary Euclidean length.

In [8: (1.15)] a *basis reduction algorithm* is given that transforms a basis $b_1, b_2, \ldots, b_j$ of a lattice L_j into a *reduced* basis $\breve{b}_1, \breve{b}_2, \ldots, \breve{b}_j$ for L_j. We won't recall the definition of a reduced basis here [8: (1.4), (1.5)], it suffices to say that the first vector $\breve{b}_1$ in such a reduced basis satisfies

(1.8) $\qquad |\underline{b}_1|^2 \le 2^{j-1}|x_j|^2$

for every $x_j \in L_j$, $x_j \neq 0$ [8: (1.11)]. The number of arithmetic operations needed by the basis reduction algorithm is $O(j^4 \log B)$, and the integers on which these operations are performed each have binary length $O(j \log B)$ [8: (1.26)].

The first time that the vector b_j is considered during the computation of a reduced basis for L_j, is at the moment that a reduced basis for L_{j-1} is obtained already; i.e. the computation of a reduced basis for L_{j-1} constitutes the first part of the computation of a reduced basis for L_j [8: (1.37)].

It follows that we can find an approximation of the shortest vector in L_n in $O(n^4 \log B)$ operations on integers having binary length $O(n \log B)$, and as a byproduct of the computation we get approximations of the shortest vectors in the lattices L_j without any time loss. If the approximation of the shortest vector in L_j, for some j, satisfies our needs already, then we break off the computation as soon as we have found this approximation, and the computation then takes $O(j^4 \log B)$ operations on integers having binary length $O(j \log B)$.

2. Factors and lattices.

This section is similar to the first part of [8: Section 2]. We formulate the generalizations of [8: (2.5),(2.6),(2.7),(2.13)] to polynomials over algebraic number fields. Let f, D, p, k, F, and H be as in Section 1. We put $n = \delta f$; we may assume that $n > 0$.

Suppose that we are given a polynomial $h \in \mathbb{Z}[\alpha][X]$ such that

(2.1) h is monic,

(2.2) $(h \bmod (p^k, H_k))$ divides $(f \bmod (p^k, H_k))$ in $W_k(\mathbb{F}_q)[X]$,

(2.3) $(h \bmod (p, H_1))$ is irreducible in $\mathbb{F}_q[X]$,

(2.4) $(h \bmod (p, H_1))^2$ does not divide $(f \bmod (p, H_1))$ in $\mathbb{F}_q[X]$.

We put $\ell = \delta h$; so $0 < \ell \leq n$. In Section 3 we will see which extra conditions have to be imposed on p so that such a polynomial h can be determined.

(2.5) Proposition. *The polynomial* f *has a unique monic irreducible factor* h_0 *in* $\frac{1}{D}\mathbb{Z}[\alpha][X]$ *for which* $(h \bmod (p, H_1))$ *divides* $(h_0 \bmod (p, H_1))$ *in* $\mathbb{F}_q[X]$. *Further, if a monic polynomial* $g \in \frac{1}{D}\mathbb{Z}[\alpha][X]$ *divides* f *in* $\mathbb{Q}(\alpha)[X]$, *then the following assertions are equivalent:*

(i) $(h \bmod (p, H_1))$ *divides* $(g \bmod (p, H_1))$ *in* $\mathbb{F}_q[X]$,

(ii) $(h \bmod (p^k, H_k))$ *divides* $(g \bmod (p^k, H_k))$ *in* $W_k(\mathbb{F}_q)[X]$,

(iii) h_0 *divides* g *in* $\mathbb{Q}(\alpha)[X]$.

In particular $(h \bmod (p^k, H_k))$ *divides* $(h_0 \bmod (p^k, H_k))$ *in* $W_k(\mathbb{F}_q)[X]$.

Proof. Use (1.1) and the proof of [8: (2.5)]. □

(2.6) In the remainder of this section we fix an integer m with $m \geq \ell$. We define L to be the collection of polynomials $g \in \frac{1}{D}\mathbb{Z}[\alpha][X]$ such that:

(i) $\delta g \leq m$,

(ii) if $\delta g = m$, then $\ell c(g) \in \mathbb{Z}$,

(iii) $(h \bmod (p^k, H_k))$ divides $(g \bmod (p^k, H_k))$ in $W_k(\mathbb{F}_q)[X]$.

We identify such a polynomial $g = \sum_{i=0}^{m-1} \sum_{j=0}^{\delta F-1} a_{ij}\alpha^j X^i + a_{m0}X^m$ (where $a_{ij} \in \frac{\mathbb{Z}}{D}$) with the $(m\delta F+1)$-dimensional vector $(a_{00}, a_{01}, \ldots, a_{0\,\delta F-1}, a_{10}, \ldots,$

$a_{m-1\,\delta F-1}, a_{m0})$. Using this identification between vectors and polynomials, it is not difficult to see that L is a lattice in $\mathbb{R}^{m\delta F+1}$; from the fact that both H and h are monic ((1.3) and (2.1)) it follows that a basis for L is given by

$$\{\tfrac{1}{D}\, p^k \alpha^j X^i : 0 \le j < \delta H,\ 0 \le i < \ell\} \cup$$

$$\{\tfrac{1}{D}\, \alpha^{j-\delta H} H(\alpha) X^i : \delta H \le j < \delta F,\ 0 \le i < \ell\} \cup$$

$$\{\tfrac{1}{D}\, \alpha^j h X^{i-\ell} : 0 \le j < \delta F,\ \ell \le i < m\} \cup$$

$$\{h X^{m-\ell}\}.$$

Notice that the matrix having these vectors as columns is upper-triangular.

We define the *length* $|g|$ of g as the ordinary Euclidean length of the vector identified with g, so $|g| = (\Sigma_{i=0}^{m-1} \Sigma_{j=0}^{\delta F-1} |a_{ij}|^2 + |a_{m0}|^2)^{\frac{1}{2}}$; the *height* g_{max} of g is defined as $\max\{|a_{ij}|\}$. Similarly we define the length and the height of polynomials in $\mathbb{Z}[T]$.

(2.7) Proposition. *Let* $b \in L$ *satisfy*

$$(2.8) \qquad p^{k\ell\delta H/\delta F} > \left(Df_{max}((n+1)\delta F(1+F_{max})^{\delta F-1})^{\frac{1}{2}}\right)^m \cdot \left(Db_{max}((m+1)\delta F(1+F_{max})^{\delta F-1})^{\frac{1}{2}}\right)^n.$$

Then b *is divisible by* h_0 *in* $\mathbb{Q}(\alpha)[X]$, *where* h_0 *is as in* (2.5). *In particular* $\gcd(f,b) \neq 1$.

The proof of this proposition is very similar to the proof of [8: (2.7)]; we therefore omit the details.

Proof. Put $g = \gcd(f,b)$, and $e = \delta g$. We may assume that g is monic. Identify the polynomials

$$(2.9)\qquad \{\alpha^j X^i f : 0 \le j < \delta F,\ 0 \le i < \delta b - e\} \cup$$

$$\{\alpha^j X^i b : 0 \le j < \delta F,\ 0 \le i < n - e\}$$

with $(\delta F(n+\delta b-e))$-dimensional vectors. The projections of these vectors on $\frac{1}{D}\mathbb{Z}X^e + \frac{1}{D}\mathbb{Z}\alpha X^e + \ldots + \frac{1}{D}\mathbb{Z}\alpha^{\delta F-1}X^e + \frac{1}{D}\mathbb{Z}X^{e+1} + \ldots + \frac{1}{D}\mathbb{Z}\alpha^{\delta F-1}X^{n+\delta b-e-1}$ form a basis for a $(\delta F(n+\delta b-2e))$-dimensional lattice M'. Using induction on j one proves that

$$(\alpha^j X^i f)_{max} = (\alpha^j f)_{max} \le f_{max}(1+F_{max})^j,$$

so that, for $0 \le j < \delta F$ and $0 \le i < \delta b - e$,

$$|\alpha^j X^i f| \le f_{max}\sqrt{(n+1)\delta F}\,(1+F_{max})^j.$$

With Hadamard's inequality, and a similar bound on $|\alpha^j X^i b|$ we get

$$d(M')^{1/\delta F} \le \left(f_{max}((n+1)\delta F(1+F_{max})^{\delta F-1})^{\frac{1}{2}}\right)^m \cdot \left(b_{max}((m+1)\delta F(1+F_{max})^{\delta F-1})^{\frac{1}{2}}\right)^n,$$

where $d(M')$ denotes the determinant of M'. With (2.8) this gives

$$(2.10)\qquad d(M') < \frac{p^{k\ell\delta H}}{D^{(n+m)\delta F}}.$$

Assume that $(h \bmod (p,H_1))$ does not divide $(g \bmod (p,H_1))$ in $\mathbb{F}_q[X]$. By Proposition (2.5) it is sufficient to derive a contradiction from this. Let $v \in \frac{1}{D}\mathbb{Z}[\alpha][X]$ be some integral linear combination of the polynomials in (2.9) such that $\delta v < e+\ell$. As in the proof of [8: (2.7)] it follows from our assumption that $(v \bmod (p^k,H_k)) = 0$ in $W_k(\mathbb{F}_q)[X]$. Therefore,

if we regard $\ell c(v)$ as a polynomial in α, we have

(2.11) $\quad \ell c(\ell c(v)) \equiv 0$ modulo p^k if $\delta\ell c(v) < \delta H$.

Now choose a basis $b_{e0}, b_{e1}, \ldots, b_{e\,\delta F-1}, b_{e+1\,0}, \ldots, b_{n+\delta b-e-1\,\delta F-1}$ for M' such that $\delta b_{ij} = i$ and $\delta\ell c(b_{ij}) = j$ for $e \le i < n+\delta b-e$ and $0 \le j < \delta F$, where $\ell c(b_{ij})$ is regarded as a polynomial in α. From (2.11) we derive that

$$\ell c(\ell c(b_{ij})) \equiv 0 \text{ modulo } p^k \quad \text{for} \quad 0 \le j < \delta H \quad \text{and} \quad e \le i < e+\ell.$$

Since $\ell c(\ell c(b_{ij})) \in \frac{\mathbb{Z}}{D}$, we obtain

$$|\ell c(\ell c(b_{ij}))| \ge \frac{p^k}{D} \quad \text{for} \quad 0 \le j < \delta H \quad \text{and} \quad e \le i < e+\ell$$

and

$$|\ell c(\ell c(b_{ij}))| \ge \frac{1}{D} \quad \text{for} \quad \delta H \le j < \delta F \quad \text{or} \quad e+\ell \le i < n+\delta b-e.$$

The determinant of M' equals the product of $|\ell c(\ell c(b_{ij}))|$, so that

$$d(M') \ge \frac{p^{k\ell\delta H}}{D^{(n+\delta b-2e)\delta F}} \ge \frac{p^{k\ell\delta H}}{D^{(n+m)\delta F}}.$$

Combined with (2.10) this is the desired contradiction. □

(2.12) To be able to formulate the generalization of [8: (2.13)] we need an upper bound on the length of monic factors of f in $\frac{1}{D}\mathbb{Z}[\alpha][X]$. In Section 4 (4.8) we prove that a monic factor of degree $\le m$ has length at most

$$f_{\max}\left(2(n+1)\delta F^3(\delta F-1)^{\delta F-1}\binom{2m}{m}\right)^{\frac{1}{2}} |F|^{2(\delta F-1)} |\mathrm{discr}(F)|^{-\frac{1}{2}},$$

where $\mathrm{discr}(F)$ denotes the discriminant of F (so $\mathrm{discr}(F) \neq 0$, since F is an irreducible polynomial in $\mathbb{Z}[T]$).

(2.13) Proposition. *Suppose that* $\tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_{m\delta F+1}$ *is a reduced basis for* L *(see* (1.7)*), and that*

$$(2.14)\qquad p^{k\ell\delta H/\delta F} > \left(2^{n(m\delta F+1)}(n+1)^{n+m}(m+1)^{n}\binom{2m}{m}^{n}\delta F^{4n+m}(\delta F-1)^{n(\delta F-1)}\right.$$
$$\left.(1+F_{max})^{(n+m)(\delta F-1)}|\mathrm{discr}(F)|^{-n}\right)^{\frac{1}{2}}\cdot(Df_{max})^{n+m}|F|^{2n(\delta F-1)}.$$

Then we have $\delta h_0 \le m$ *if and only if* (2.8) *is satisfied with* b *replaced by* $\tilde{b}_1$.

Proof. Use (2.12), (1.8), and the proof of [8: (2.13)]. □

3. Description of the algorithm.

Let f be a polynomial in $\mathbb{Q}(\alpha)[X]$ of degree n, with $n > 0$. We describe an algorithm to compute the irreducible factors of f in $\mathbb{Q}(\alpha)[X]$.

For the moment we assume that f is monic. If D, p, H, and h are chosen in such a way that the conditions in Sections 1 and 2 are satisfied, then we can determine the factor h_0 of f by means of Propositions (2.7) and (2.13); this is described in more detail in Algorithm (3.1). After that, we explain in (3.4) how we choose D, p, H, and h, and we analyze the running time of the resulting factorization algorithm.

(3.1) Suppose that a positive integer D, a prime number p, and polynomials $H \in \mathbb{Z}[T]$ and $h \in \mathbb{Z}[\alpha][X]$ are given such that (1.1), (1.2), (1.3), (1.5), (1.6), (2.1), (2.3), and (2.4), and (1.4) and (2.2) with k replaced by 1, are satisfied. We describe an algorithm that determines h_0, the monic irreducible factor of f for which $(h \bmod (p,H_1))$ divides $(h_0 \bmod (p,H_1))$, cf. (2.5).

Put $\ell = \delta h$; we may assume that $\ell < n$. We calculate the least positive integer k for which (2.14) holds with m replaced by $n-1$:

$$(3.2)\qquad p^{k\ell\delta H/\delta F} > \Big(2^{n((n-1)\delta F+1)}(n+1)^{2n-1}\, n^n \binom{2(n-1)}{n-1}^n \delta F^{5n-1}(\delta F-1)^{n(\delta F-1)} (1+F_{max})^{(2n-1)(\delta F-1)} |\mathrm{discr}(F)|^{-n}\Big)^{\frac{1}{2}} \cdot (Df_{max})^{2n-1}|F|^{2n(\delta F-1)}.$$

Next we modify H in such a way that (1.4) holds for the value of k just calculated. The factor $H_k = (H \bmod p^k)$ of $(F \bmod p^k)$ gives us the possibility to compute in $W_k(\mathbb{F}_q)$. Therefore we now modify h, without changing $(h \bmod (p, H_1))$, in such a way that (2.2) holds for the above value of k. The computations of the new H and h can both be done by means of Hensel's lemma [5: exercise 4.6.22; 14; 13]; notice that Hensel's lemma can be applied because of (1.6) and (2.4).

Now apply the basis reduction algorithm [8: (1.15)] to the $(m\delta F+1)$-dimensional lattice L as defined in (2.6), for each of the values $m = \ell, \ell+1, \ldots, n-1$ in succession; but we stop as soon as for one of these values of m we find a basis $\tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_{m\delta F+1}$ for L such that (2.8) is satisfied with b replaced by $\tilde{b}_1$. If such a basis is found for a certain value m_0 of m, then we know from (2.13) that $\delta h_0 \le m_0$. Since we try the values $m = \ell, \ell+1, \ldots, n-1$ in succession we also know from (2.13) that $\delta h_0 > m_0 - 1$, so $\delta h_0 = m_0$. By (2.7) the polynomial h_0 divides $\tilde{b}_1$ in $\mathbb{Q}(\alpha)[X]$ which implies, together with $\delta \tilde{b}_1 \le m_0$, that $\delta \tilde{b}_1 = m_0$. From (2.6)(ii) and from the fact that h_0 is monic we find that $\tilde{b}_1 = ch_0$, for some constant $c \in \mathbb{Z}$. Using that $h_0 \in L$ and that $\tilde{b}_1$ belongs to a basis for L, we conclude that $c = \pm 1$, so that $\tilde{b}_1 = \pm h_0$.

If on the other hand we did not find such a basis for L, then we know from (2.13) that $\delta h_0 > n-1$. This implies that $h_0 = f$. This finishes the

description of Algorithm (3.1).

(3.3) Proposition. *Denote by* $m_0 = \delta h_0$ *the degree of the irreducible factor* h_0 *of* f *that is found by Algorithm* (3.1). *Then the number of arithmetic operations needed by Algorithm* (3.1) *is* $O(m_0(n^5\delta F^6 + n^4\delta F^6 \log(\delta F|F|) + n^4\delta F^5 \log(Df_{max}) + n^3\delta F^4 \log p))$ *and the integers on which these operations are performed each have binary length* $O(n^3\delta F^3 + n^2\delta F^3 \log(\delta F|F|) + n^2\delta F^2 \log(Df_{max}) + n\delta F \log p)$.

Proof. Let m_1 be the largest value of m for which the basis reduction algorithm is performed, so $m_1 = m_0$ or $m_1 = m_0 - 1$. From (1.7) it follows that during the computation of the reduced basis for the $(m_1\delta F+1)$-dimensional lattice, also reduced bases were obtained for the $(m\delta F+1)$-dimensional lattices, for $\ell \le m < m_1$. Therefore the number of arithmetic operations needed for the applications of the basis reduction algorithm is $O((m_1\delta F)^4 \log B)$, and the integers on which these operations are performed each have binary length $O(m_1\delta F \log B)$, where B bounds the length of the vectors in the initial basis for L (cf. (2.6)). Assuming that the coefficients of the initial basis are reduced modulo p^k, we derive from (3.2), $|\mathrm{discr}(F)| \ge 1$, $\delta H \ge 1$, and $\ell \ge 1$ that

$$\log B = O(n^2\delta F^2 + n\delta F^2 \log(\delta F|F|) + n\delta F \log(Df_{max}) + \log p).$$

Combined with $m_1 = O(m_0)$ this yields the estimates given in (3.3).

It is straightforward to verify that the same estimates are valid for both applications of Hensel's lemma and for the computation of $\mathrm{discr}(F)$.

□

(3.4) We now describe how to choose D, p, H, and h in such a way that Algorithm (3.1) can be applied. The algorithm to factor f into its monic irreducible factors in $\mathbb{Q}(\alpha)[X]$ then easily follows.

First we choose a positive integer D such that (1.1) holds, i.e. f and all monic factors of f in $\mathbb{Q}(\alpha)[X]$ are in $\frac{1}{D}\mathbb{Z}[\alpha][X]$. From [14] it follows that we can take $D = dc$, where d is such that $f \in \frac{1}{d}\mathbb{Z}[\alpha][X]$, and c is the largest integer such that c^2 divides $\mathrm{discr}(F)$. This integer c however might be difficult to compute; therefore we take $D = d\,|\mathrm{discr}(F)|$ as denominator, which clearly also suffices.

We may assume that the resultant $R(f,f') \in \mathbb{Q}(\alpha)$ of f and its derivative f' is unequal to zero, i.e. f has no multiple factors in $\mathbb{Q}(\alpha)[X]$. We apply the algorithm from [10] to determine p as the smallest prime number not dividing $D \cdot \mathrm{discr}(F) \cdot R(f,f')$; so (1.2) is satisfied.

Using Berlekamp's algorithm [5: Section 4.6.2] we compute the irreducible factorization $(F \bmod p) = \prod_{i=1}^{t} (G_i \bmod p)$ of $(F \bmod p)$ in $(\mathbb{Z}/p\mathbb{Z})[T]$. This factorization does not contain multiple factors because $\mathrm{discr}(F) \not\equiv 0$ modulo p. Combined with $R(f,f') \not\equiv 0$ modulo p this implies that there exists an integer $i_0 \in \{1,2,\ldots,t\}$ such that

$$(R(f,f') \bmod (p, (G_{i_0} \bmod p))) \not\equiv 0;$$

Let H be such a polynomial G_{i_0}. We may assume that H is monic, so that (1.3), (1.5), (1.6), and (1.4) with k replaced by 1 are satisfied.

Next we determine the irreducible factorization of $(f \bmod (p, H_1))$ in $\mathbb{F}_q[X]$ by means of Berlekamp's algorithm [2: Section 5], where $q = p^{\delta H}$ and $\mathbb{F}_q \simeq (\mathbb{Z}/p\mathbb{Z})[T]/(H \bmod p)$. (Notice that we use a modified version of Berlekamp's algorithm here, one that is polynomial-time in p and δH rather than polynomial-time in the number of elements of the finite field.)

Since f is monic the resultant $R(f,f')$ is, up to sign, equal to the discriminant of f, so that it follows from the construction of H that the discriminant of f is unequal to zero in $\mathbf{F}_q$. Therefore (2.4) holds for all irreducible factors $(h \bmod (p,H_1))$ of $(f \bmod (p,H_1))$ in $\mathbf{F}_q[X]$; we may assume that these factors are monic.

The algorithm to factorize f now follows by repeated application of Algorithm (3.1).

(3.5) Proposition. *The algorithm sketched above computes the irreducible factorization of any monic polynomial* $f \in \frac{1}{d}\mathbb{Z}[\alpha][X]$ *of degree* $n > 0$. *The number of arithmetic operations needed by the algorithm is* $O(n^6\delta F^6 + n^5\delta F^6 \log(\delta F|F|) + n^5\delta F^5 \log(df_{max}))$, *and the integers on which these operations are performed each have binary length* $O(n^3\delta F^3 + n^2\delta F^3 \log(\delta F|F|) + n^2\delta F^2 \log(df_{max}))$.

Proof. It follows from [3] that the calculations of $R(f,f')$ and $\mathrm{discr}(F)$ satisfy the above estimates. From Hadamard's inequality we obtain

$$|\mathrm{discr}(F)| \leq \delta F^{\delta F} |F|^{2\delta F-1};$$

it follows that

$$\log D = O(\log d + \delta F \log(\delta F|F|)).$$

Let A be a matrix having entries $A_{ij} = \sum_{\ell=0}^{\delta F-1} a_{ij\ell} T^{\ell} \in \mathbb{Z}[T]$, for $1 \leq i,j \leq m$, and some positive integer m. The determinant $d(A)$ of A is a polynomial of degree $\leq m(\delta F-1)$ in $\mathbb{Z}[T]$. According to [4] the length, and therefore the height, of $d(A)$ is bounded from above by

$$\left(\prod_{j=1}^{m} \sum_{i=1}^{m} \left(\sum_{\ell=0}^{\delta F-1} |a_{ij\ell}|\right)^2\right)^{\frac{1}{2}}.$$

Using this bound it is easily proven that the height of $d(A)$ modulo F is bounded by

$$\left(\Pi_{j=1}^{m} \Sigma_{i=1}^{m} (\Sigma_{\ell=0}^{\delta F-1} |a_{ij\ell}|)^2\right)^{\frac{1}{2}} (1+F_{max})^{(m-1)(\delta F-1)}.$$

It follows that

$$(R(f,f'))_{max} \leq (\sqrt{n+1}\delta F f_{max})^{n-1} (\sqrt{n}\delta F n f_{max})^{n} (1+F_{max})^{(2n-2)(\delta F-1)},$$

where $R(f,f')$ is regarded as a polynomial in α. We find from the definitions of D and p that

$$\Pi_{q \text{ prime}, q < p} q \leq d \cdot \mathrm{discr}(F) \cdot (R(df,df'))_{max}$$

and this yields in a similar way as in [8] that

$$p = O(\log d + n\delta F \log(\delta F|F|) + n \log n + n \log(df_{max})).$$

This implies that the computation of the prime number p, and the computation of the factorizations of $(F \bmod p)$ in $(\mathbb{Z}/p\mathbb{Z})[T]$ and $(f \bmod (p,H_1))$ in $\mathbf{F}_q[X]$ satisfy the estimates in (3.5). Proposition (3.5) now easily follows from the bounds on $\log D$ and p, and from the observation that a monic factor g of f in $\mathbb{Q}(\alpha)[X]$ satisfies $\log(g_{max}) = O(\delta F \log(\delta F|F|) + n + \log(f_{max}))$ (see (4.7)). □

(3.6) We now drop the assumption that f is monic, so let f be a polynomial of degree $n > 0$ in $\mathbb{Z}[\alpha][X]$. We show that there exists a monic polynomial $\tilde{f} = \ell c(f)^{-1} f \in \frac{1}{d}\mathbb{Z}[\alpha][X]$, such that $\log(d\tilde{f}_{max}) = O(\delta F \log(\delta F|F|) + \delta F \log(f_{max}))$, for some non-zero integer d.

Denote by $C(\alpha) = \Sigma_{i=0}^{\delta F-1} C_i \alpha^i \in \mathbb{Z}[\alpha]$ the leading coefficient of f. The resultant $R(C,F) \in \mathbb{Z}$ of C and F is defined as the determinant of

the following matrix:

$$\begin{pmatrix} C_0 & 0 & \cdots & 0 & F_0 & 0 & \cdots & 0 \\ \vdots & \ddots & & \vdots & \vdots & \ddots & & \vdots \\ & & & 0 & & & & 0 \\ & & & C_0 & & & & F_0 \\ C_{\delta F-1} & & & & F_{\delta F} & & & \\ 0 & \ddots & & & 0 & \ddots & & \\ \vdots & & & & \vdots & & & \\ 0 & \cdots & 0 & C_{\delta F-1} & 0 & \cdots & 0 & F_{\delta F} \end{pmatrix}$$

$\longleftarrow \delta F \longrightarrow \quad \longleftarrow \delta F-1 \longrightarrow$

where $F(T) = \Sigma_{i=0}^{\delta F} F_i T^i$. We add, for $2 \le i \le 2\delta F-1$, the i-th row times T^{i-1} to the first row, so that the first row of the matrix becomes $(C(T), TC(T), \ldots, T^{\delta F-1}C(T), F(T), TF(T), \ldots, T^{\delta F-2}F(T))$. Expanding the determinant of the resulting matrix with respect to the first row gives

$$R(C,F) = C(T)\cdot(R_{\delta F-1}T^{\delta F-1}+\ldots+R_1T+R_0) + F(T)\cdot(S_{\delta F-2}T^{\delta F-2}+\ldots+S_1T+S_0),$$

where $R_i, S_j \in \mathbb{Z}$ for $0 \le i < \delta F$ and $0 \le j < \delta F-1$.

The values R_i and S_j are determinants of $(2\delta F-2)\times(2\delta F-2)$ submatrices of the above matrix, and therefore, using Hadamard's inequality, $|R_i|$ and $|S_j|$ are both bounded from above by

$$(\sqrt{\delta F}\,|F|f_{max})^{\delta F}.$$

The evaluation of these determinants can be done by means of the methods described in [1]. Putting $R(T) = \Sigma_{i=0}^{\delta F-1} R_i T^i$ and $d = R(C,F)$ we find that $C(T)R(T) \equiv d \bmod F(T)$, so that $\frac{R(\alpha)}{d} \in \frac{1}{d}\mathbb{Z}[\alpha]$ is the inverse of $C(\alpha)$. Now use Hadamard's inequality to derive an upper bound for d, and we find that the monic polynomial $\tilde{f} = \frac{R(\alpha)}{d} f \in \frac{1}{d}\mathbb{Z}[\alpha][X]$ satisfies the estimates given above.

(3.7) Theorem. *Let* f *be a polynomial of degree* $n > 0$ *in* $\mathbb{Z}[\alpha][X]$. *The irreducible factorization of* f *in* $\mathbb{Q}(\alpha)[X]$ *can be computed in* $O(\delta F^6(n^6 + n^5\log(\delta F|F|) + n^5\log(f_{max})))$ *arithmetic operations on integers having binary length* $O(\delta F^3(n^3 + n^2\log(\delta F|F|) + n^2\log(f_{max})))$.

Proof. The proof follows from (3.6) and (3.5). □

4. Coefficient bound for factors.

We use the method sketched in [14] to derive an explicit upper bound for the height and the length of a monic divisor of a monic polynomial in $\mathbb{Q}(\alpha)[X]$.

For polynomials in $\mathbb{Q}(\alpha)[X]$ the height and the length are defined as in (2.6); for a polynomial $g = \Sigma_i\, c_i X^i \in \mathbb{C}[X]$, where $\mathbb{C}$ denotes the complex numbers, the length $|g|$ is defined as $(\Sigma_i |c_i|^2)^{\frac{1}{2}}$.

Let $\alpha_1, \alpha_2, \ldots, \alpha_{\delta F}$ denote the conjugates of α, i.e. $\alpha_1, \alpha_2, \ldots, \alpha_{\delta F} \in \mathbb{C}$ are the roots of the minimal polynomial F. For an element $\beta = \Sigma_{i=0}^{\delta F-1} b_i \alpha^i \in \mathbb{Q}(\alpha)$ the conjugates of β are defined as $\Sigma_{i=0}^{\delta F-1} b_i \alpha_j^i$ for $1 \le j \le \delta F$. We define $\|\beta\| \in \mathbb{R}$ as the largest absolute value of any of the conjugates of β; so $\|\alpha\|$ is the largest absolute value of any of the roots of F.

For any choice of $\alpha \in \{\alpha_1, \alpha_2, \ldots, \alpha_{\delta F}\}$ a polynomial $g \in \mathbb{Q}(\alpha)[X]$ can be regarded as a polynomial $\Sigma_{i=0}^{\delta g} c_{ji} X^i \in \mathbb{C}[X]$; we define $\|g\|$ as $\max_{1 \le j \le \delta F} \{|\Sigma_{i=0}^{\delta g} c_{ji} X^i|\}$.

Now let $f \in \mathbb{Q}(\alpha)[X]$ be a monic polynomial of degree n, and let $g = \Sigma_{i=0}^{m} g_i X^i \in \mathbb{Q}(\alpha)[X]$ be a monic factor of degree m of f. Since both f and g are monic, we obtain from [9] that

(4.1) $\quad \|g_i\| \le \binom{m}{i} \|f\|, \quad \text{for} \quad 0 \le i \le m.$

From (4.1) we will derive bounds on the height and the length of g.

Let $S = (s_{ij})_{i,j=0}^{\delta F-1}$ be the $\delta F \times \delta F$ matrix with $s_{ij} = \alpha_{j+1}^{i}$. Since S is a Vandermonde matrix and because the roots of F are distinct, it follows that S is invertible, and that the absolute value of the determinant of S equals $|\mathrm{discr}(F)|^{\frac{1}{2}}$. We denote by $T = (t_{ij})_{i,j=0}^{\delta F-1}$ the matrix S^{-1}, and by $|T| = \max\{\Sigma_{i=0}^{\delta F-1} |t_{ij}| : 0 \le j < \delta F\}$ (this is the L_∞-norm for matrices).

Let $r_j \in \mathbb{C}$ be the conjugates of $g_i = \Sigma_{k=0}^{\delta F-1} g_{ik}\alpha^k \in \mathbb{Q}(\alpha)$, for $1 \le j \le \delta F$, then we have

$$(g_{i0}, g_{i1}, \ldots, g_{i\,\delta F-1}) \cdot S = (r_1, r_2, \ldots, r_{\delta F}),$$

and therefore

(4.2) $\quad (g_{i0}, g_{i1}, \ldots, g_{i\,\delta F-1}) = (r_1, r_2, \ldots, r_{\delta F}) \cdot T$

for $0 \le i \le m$. From (4.1) we have that

$$|r_j| \le \binom{m}{i} \|f\| \quad \text{for} \quad 1 \le j \le \delta F$$

and this gives, combined with (4.2),

$$|g_{ik}| \le \binom{m}{i} |T| \, \|f\| \quad \text{for} \quad 1 \le k < \delta F \quad \text{and} \quad 0 \le i \le m.$$

This implies that

(4.3) $\quad g_{\max} \le \binom{m}{m/2} |T| \, \|f\|,$

and

$$(4.4)\qquad |g| = (\Sigma_{i=0}^{m} \Sigma_{k=0}^{\delta F-1} g_{ik}^2)^{\frac{1}{2}} \le \left(\delta F \Sigma_{i=0}^{m} \binom{m}{i}^2\right)^{\frac{1}{2}} |T| \ \|f\|$$

$$= \left(\delta F \binom{2m}{m}\right)^{\frac{1}{2}} |T| \ \|f\|.$$

It remains to give upper bounds for $|T|$ and $\|f\|$.

The entries of T are determinants of $(\delta F-1)\times(\delta F-1)$ submatrices of S, divided by $|\mathrm{discr}(F)|^{\frac{1}{2}}$. Using Hadamard's inequality we get the upper bound

$$\Pi_{j=1}^{\delta F-1} (\Sigma_{i=1}^{\delta F-1} |\alpha_j|^{2i})^{\frac{1}{2}}$$

for the determinant of such a $(\delta F-1)\times(\delta F-1)$ submatrix of S. This easily yields the bound

$$\Pi_{|\alpha_j| \le 1} |\alpha_j| \cdot \Pi_{|\alpha_j| > 1} |\alpha_j|^{(\delta F-1)^{\frac{1}{2}}} |\alpha_j|^{\delta F-1}$$

$$= (\delta F-1)^{(\delta F-1)/2} (\Pi_{|\alpha_j| > 1} |\alpha_j|)^{\delta F-1}$$

Since F is monic we know from [9: Theorem 2] that $\Pi_{|\alpha_j| > 1} |\alpha_j| \le |F|$, so that we arrive at the bound

$$(\delta F-1)^{(\delta F-1)/2} |F|^{\delta F-1} |\mathrm{discr}(F)|^{-\frac{1}{2}}$$

for the absolute values of the entries of T. It follows that

$$(4.5)\qquad |T| \le \delta F (\delta F-1)^{(\delta F-1)/2} |F|^{\delta F-1} |\mathrm{discr}(F)|^{-\frac{1}{2}}.$$

A straightforward computation yields the bound

$$\|f\| \le \max_{1 \le j \le \delta F} (\Sigma_{i=0}^{n} f_{max}^2 \Sigma_{k=0}^{\delta F-1} |\alpha_j|^{2k})^{\frac{1}{2}}$$

$$\le \sqrt{n+1} (\Sigma_{k=0}^{\delta F-1} \|\alpha\|^{2k})^{\frac{1}{2}} f_{max}.$$

There are several easily calculated upper bounds for $\|\alpha\|$, for instance $\|\alpha\| \leq 1 + F_{max}$ and $\|\alpha\| \leq |F|$ (cf. [11]). For simplicity we take $\|\alpha\| \leq |F|$, so that we obtain

$$(4.6) \qquad \|f\| \leq \sqrt{n+1}\,(\Sigma_{k=0}^{\delta F-1} |F|^{2k})^{\frac{1}{2}} f_{max} = \sqrt{n+1}\left(\frac{|F|^{2\delta F}-1}{|F|^2-1}\right)^{\frac{1}{2}} f_{max}$$
$$\leq \sqrt{n+1}\left(\frac{|F|^{2\delta F}}{\frac{1}{2}|F|^2}\right)^{\frac{1}{2}} f_{max} = \sqrt{n+1}\sqrt{2}\,|F|^{\delta F-1} f_{max}.$$

Combining (4.3), (4.4), (4.5), and (4.6) we finally get

$$(4.7) \qquad g_{max} \leq f_{max}\left(2(n+1)(\delta F-1)^{\delta F-1}\right)^{\frac{1}{2}} |F|^{2(\delta F-1)} \delta F \binom{m}{m/2} |\mathrm{discr}(F)|^{-\frac{1}{2}}$$

and

$$(4.8) \qquad |g| \leq f_{max}\left(2(n+1)\delta F^3(\delta F-1)^{\delta F-1}\binom{2m}{m}\right)^{\frac{1}{2}} |F|^{2(\delta F-1)} |\mathrm{discr}(F)|^{-\frac{1}{2}}.$$

Acknowledgements.

Acknowledgements are due to H.W. Lenstra, Jr. and P. van Emde Boas for several helpful remarks.

References.

1. E.H. Bareiss, Sylvester's identity and multistep integer-preserving Gaussian elimination, Math. Comp. 22 (1968), 565-578.
2. E.R. Berlekamp, Factoring polynomials over large finite fields, Math. Comp. 24 (1970), 713-735.
3. W.S. Brown, The subresultant PRS algorithm, ACM Transactions on mathematical software 4 (1978), 237-249.
4. A.J. Goldstein, R.L. Graham, A Hadamard-type bound on the coefficients of a determinant of polynomials, SIAM Rev. 16 (1974), 394-395.

5. D.E. Knuth, The art of computer programming, vol. 2, Seminumerical algorithms, Addison-Wesley, Reading, second edition 1981.

6. S. Landau, Factoring polynomials over algebraic number fields is in polynomial time, unpublished manuscript.

7. A.K. Lenstra, Lattices and factorization of polynomials over algebraic number fields, Proceedings Eurocam '82, LNCS 144, 32-39.

8. A.K. Lenstra, H.W. Lenstra, Jr., L. Lovász, Factoring polynomials with rational coefficients, Report IW 195/82, Mathematisch Centrum, Amsterdam 1982.

9. M. Mignotte, An inequality about factors of polynomials, Math. Comp. 28 (1974), 1153-1157.

10. P. Pritchard, A sublinear additive sieve for finding prime numbers, Comm. ACM 24 (1981), 18-23.

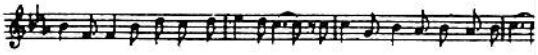
11. J. Stoer, Einführung in die numerische Mathematik I, Springer, Berlin 1972.

12. B.M. Trager, Algebraic factoring and rational function integration, Proceedings of the 1976 ACM symposium on symbolic and algebraic computation, 219-226.

13. P.S. Wang, Factoring multivariate polynomials over algebraic number fields, Math. Comp. 30 (1976), 324-336.

14. P.J. Weinberger, L.P. Rothschild, Factoring polynomials over algebraic number fields, ACM Transactions on mathematical software 2 (1976), 335-350.

15. D.Y.Y. Yun, The Hensel lemma in algebraic manipulation, MIT, Cambridge 1974; reprint: Garland Publ. Co., New York 1980.



Factoring multivariate polynomials over finite fields

by

A.K. Lenstra

ABSTRACT

This paper describes an algorithm for the factorization of multivariate polynomials with coefficients in a finite field that is polynomial-time in the degrees of the polynomial to be factored. The algorithm makes use of a new basis reduction algorithm for lattices over $\mathbb{F}_q[Y]$.

KEY WORDS & PHRASES: *polynomial algorithm, polynomial factorization*

Factoring multivariate polynomials over finite fields.

We present an algorithm for the factorization of multivariate polynomials with coefficients in a finite field. Let f be a polynomial in $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$ of degree n_i in X_i, where $\mathbb{F}_q$ denotes a finite field containing q elements, for some prime power $q = p^m$. To factor f, our algorithm needs a number of arithmetic operations in $\mathbb{F}_q$ that is bounded by a polynomial function of $\prod_{i=1}^{t} n_i$ and $p\,m$.

If the number of variables t equals two, then our algorithm is similar to the polynomial-time algorithm for the factorization of polynomials in one variable with rational coefficients [7]. An outline of the algorithm to factor $f \in \mathbb{F}_q[X, Y]$ is as follows. For a suitably chosen irreducible polynomial $F \in \mathbb{F}_q[Y]$, and a large enough positive integer k, we determine a factor h of f modulo the ideal (F^k). The irreducible factor h_0 of f for which h divides h_0 modulo (F^k) can be regarded as an element of a certain lattice over $\mathbb{F}_q[Y]$. We prove that h_0 is, in a certain sense, the shortest element in this lattice, and we show that this enables us to determine h_0 by means of a new basis reduction algorithm for lattices over $\mathbb{F}_q[Y]$.

For $f \in \mathbb{F}_q[X_1, X_2, \ldots, X_t]$ with $t > 2$, we first substitute high enough powers of X_2 for X_3 up to X_t. We then proceed in a similar way as above with the resulting polynomial in $\mathbb{F}_q[X_1, X_2]$.

The basis reduction algorithm for lattices over $\mathbb{F}_q[Y]$ is described in Section 1. If we define the norm of a vector over $\mathbb{F}_q[Y]$ as its degree in Y, then this algorithm enables us to determine the successive minima of a lattice over $\mathbb{F}_q[Y]$.

The algorithm to factor polynomials in $\mathbb{F}_q[X, Y]$ is presented in Section 2; the results are similar to Section 2 and 3 of [7]. In Section 3 the algorithm for polynomials in more than two variables over a finite field is explained.

Other recent publications on this subject are [5] and [6]. For two variables the algorithm from [5] is similar to ours; it only differs in the determination of short vectors in a lattice over $\mathbb{F}_q[Y]$. Also the generalization to more than two variables is distinct from ours. Another approach is given in [6].

1. The reduction algorithm.

Let n be a positive integer, and let $\mathbb{F}_q$ denote the finite field containing q elements, for some prime power q. For a rational function $g \in \mathbb{F}_q(Y)$ we denote by $|g|$ its degree in Y (i.e. the degree of the numerator minus the degree of the denominator); we put $|0| = -\infty$. The *norm* $|a|$ of an n-dimensional vector $a = (a_1, a_2, \ldots, a_n) \in \mathbb{F}_q(Y)^n$ is defined as $\max\{|a_i| : 1 \le i \le n\}$.

Let $b_1, b_2, \ldots, b_n \in \mathbb{F}_q[Y]^n \subset \mathbb{F}_q(Y)^n$ be linearly independent over $\mathbb{F}_q[Y]$; we denote by $b_{ij} \in \mathbb{F}_q[Y]$ the j-th coordinate of b_i. The *lattice* $L \subset \mathbb{F}_q[Y]^n$ of *rank* n spanned by $b_1, b_2, \ldots, b_n$ is defined as

$$L = \Sigma_{i=1}^{n} \mathbb{F}_q[Y]\, b_i = \{\Sigma_{i=1}^{n} r_i b_i : r_i \in \mathbb{F}_q[Y] \quad (1 \le i \le n)\}.$$

The *determinant* $d(L) \in \mathbb{F}_q[Y]$ of L is defined as the determinant of the $n \times n$ matrix B having the vectors $b_1, b_2, \ldots, b_n$ as rows. It is well-known that, up to units in $\mathbb{F}_q$, the value of $d(L)$ does not depend on the choice of basis for L. The *orthogonality defect* $OD(b_1, b_2, \ldots, b_n)$ of a basis $b_1, b_2, \ldots, b_n$ for a lattice L is defined as $\Sigma_{i=1}^{n} |b_i| - |d(L)|$. Clearly $OD(b_1, b_2, \ldots, b_n) \ge 0$.

(1.1) Proposition. Let $x = \Sigma_{i=1}^{n} r_i b_i \in L$. Then

$$|r_i b_i| \le |x| + OD(b_1, b_2, \ldots, b_n)$$

for $1 \le i \le n$.

Proof. The norm of the i-th column of B^{-1} is bounded from above by $\Sigma_{j=1}^{n} |b_j| - |b_i| - |d(L)| = OD(b_1, b_2, \ldots, b_n) - |b_i|$ by Cramer's rule. Since r_i is the inner product of x and the i-th column of B^{-1}, we have

that $|r_i| \le |x| + OD(b_1, b_2, \ldots, b_n) - |b_i|$, which proves (1.1). □

For $1 \le j \le n$ a *j-th successive minimum* $|m_j|$ of L is recursively defined as the norm of a vector of smallest norm in L that is linearly independent of $m_1, m_2, \ldots, m_{j-1}$ over $\mathbb{F}_q[Y]$. It is well-known that $|m_j|$ is independent of the particular choice of $m_1, m_2, \ldots, m_{j-1}$ (cf. [8]).

(1.2) Proposition. Let $b_1, b_2, \ldots, b_n$ be a basis for a lattice L satisfying $OD(b_1, b_2, \ldots, b_n) = 0$, ordered in such a way that $|b_i| \le |b_j|$ for $1 \le i < j \le n$. Then $|b_j|$ is a j-th successive minimum of L for $1 \le j \le n$, and in particular $|b_1| \le |x|$ for every $x \in L$, $x \neq 0$.

Proof. Let $|x|$ be a j-th successive minimum of L, for some j, $1 \le j \le n$. It is sufficient to prove that $|x| \ge |b_j|$. Suppose that $x = \sum_{i=1}^{n} r_i b_i$. Clearly there must be an index $i_0 \in \{j, j+1, \ldots, n\}$ such that $r_{i_0} \neq 0$. Proposition (1.1) yields that

$$|x| \ge |r_{i_0} b_{i_0}| \ge |b_{i_0}| \ge |b_j|,$$

which proves (1.2). □

We say that the basis $b_1, b_2, \ldots, b_n$ is *reduced* if the columns of B (i.e. the coordinates of the vectors $b_1, b_2, \ldots, b_n$) can be permuted in such a way that the rows $\bar{b}_1, \bar{b}_2, \ldots, \bar{b}_n$ of the resulting matrix satisfy

(1.3) $\quad |\bar{b}_i| \le |\bar{b}_j|$ for $1 \le i < j \le n$,

(1.4) $\quad |\bar{b}_{ii}| \ge |\bar{b}_{ij}|$ for $1 \le i < j \le n$,

(1.5) $\quad |\bar{b}_{ii}| > |\bar{b}_{ij}|$ for $1 \le j < i \le n$.

Conditions (1.4) and (1.5) are illustrated in Figure 1; observe that $|b_i| = |\overline{b}_i|$.

$$\begin{pmatrix} = |b_1| & \leq |b_1| & \leq |b_1| & . & . & . & \leq |b_1| \\ < |b_2| & = |b_2| & \leq |b_2| & . & . & . & \leq |b_2| \\ < |b_3| & < |b_3| & = |b_3| & . & . & . & \leq |b_3| \\ \vdots & \vdots & \vdots & & & & \vdots \\ < |b_n| & < |b_n| & < |b_n| & . & . & . & = |b_n| \end{pmatrix}$$

Figure 1. The j-th position in the i-th row gives the condition that holds for $|\overline{b}_{ij}|$ if $b_1, b_2, \ldots, b_n$ is a reduced basis.

(1.6) Remark. It follows from (1.4) and (1.5) that a reduced basis $b_1, b_2, \ldots, b_n$ for a lattice L satisfies $OD(b_1, b_2, \ldots, b_n) = 0$. Combined with (1.3) and (1.2) this implies that $|b_j|$ is a j-th successive minimum of L, for $1 \leq j \leq n$, and b_1 is a shortest vector in L.

(1.7) We now describe an algorithm that transforms a basis $b_1, b_2, \ldots, b_n$ for a lattice L into a reduced basis for L. In the course of this algorithm the coordinates of $b_1, b_2, \ldots, b_n$ will be permuted in such a way that at the end of the algorithm (1.3), (1.4), and (1.5) hold with $\overline{b}_1, \overline{b}_2, \ldots, \overline{b}_n$ replaced by $b_1, b_2, \ldots, b_n$; the original ordering of the coordinates can then be restored by applying the appropriate inverse permutation of the coordinates. For simplicity we take $|b_0| = -\infty$.

Suppose that an integer $k \in \{0, 1, \ldots, n\}$ is given such that

(1.8) $\quad |b_i| \leq |b_j|$ for $1 \leq i < j \leq k$,

(1.9) $\quad |b_k| \leq |b_j|$ for $k < j \leq n$,

(1.10) $\quad |b_{ii}| \geq |b_{ij}|$ for $1 \leq i \leq k$ and $i < j \leq n$,

(1.11) $\quad |b_{ii}| > |b_{ij}|$ for $1 \leq j < i \leq k$.

(Initially these conditions are satisfied for $k = 0$.) In this situation we proceed as follows. If $k = n$, then the basis is reduced, and the algorithm terminates. Suppose that $k < n$. Renumber $\{b_{k+1}, b_{k+2}, \ldots, b_n\}$ in such a way that $|b_{k+1}| = \min\{|b_i|: k+1 \leq i \leq n\}$. Let $a_{ij} \in \mathbb{F}_q$ be the coefficient of $Y^{|b_i|}$ in b_{ij} for $1 \leq i \leq k+1$ and $1 \leq j \leq k$. It follows from (1.10) and (1.11) that $a_{ii} \neq 0$ for $1 \leq i \leq k$, and that $a_{ij} = 0$ for $1 \leq j < i \leq k$. This implies that a solution $(r_1, r_2, \ldots, r_k)$, with $r_i \in \mathbb{F}_q$, of the following triangular system of equations over $\mathbb{F}_q$ exists:

(1.12) $\quad \Sigma_{i=1}^{k} a_{ij} r_i = a_{k+1\,j}$ for $1 \leq j \leq k$.

We put

(1.13) $\quad b_{k+1}^{*} = b_{k+1} - \Sigma_{i=1}^{k} r_i b_i Y^{|b_{k+1}| - |b_i|}$,

then $|b_{k+1}^{*}| \leq |b_{k+1}|$, and, with (1.8) and (1.9), $b_{k+1}^{*} \in \mathbb{F}_q[Y]^n$. Furthermore, (1.12) implies that $|b_{k+1\,i}^{*}| < |b_{k+1}|$ for $1 \leq i \leq k$. We distinguish two cases.

If $|b_{k+1}^{*}| = |b_{k+1}|$, then we replace b_{k+1} by b_{k+1}^{*}, we permute the coordinates of $b_1, b_2, \ldots, b_n$ in such a way that $|b_{k+1\,k+1}| = |b_{k+1}|$ (this does not affect the first k coordinates), and finally we replace k by $k+1$.

If, on the other hand, $|b_{k+1}^{*}| < |b_{k+1}|$, then we replace b_{k+1} by b_{k+1}^{*} and we replace k by the largest index $\ell \in \{0, 1, \ldots, k\}$ such that $|b_\ell| \leq |b_{k+1}|$.

We are now in the situation as described in (1.8), (1.9), (1.10), and (1.11), and we proceed with the algorithm from there. This finishes the description of Algorithm (1.7).

We shall now analyze the running time of Algorithm (1.7). By an *arithmetic operation in* $\mathbb{F}_q$ we mean an addition, subtraction, multiplication or division of two elements of $\mathbb{F}_q$.

(1.14) Proposition. Algorithm (1.7) takes $O(n^3 B (OD(b_1, b_2, \ldots, b_n) + 1))$ arithmetic operations in $\mathbb{F}_q$ to transform a basis $b_1, b_2, \ldots, b_n$ for a lattice L into a reduced basis for L, where $B \in \mathbb{Z}_{\geq 2}$ is chosen in such a way that $|b_i| \leq B$ for $1 \leq i \leq n$.

Proof. To prove that Algorithm (1.7) terminates, consider $S = \Sigma_{i=1}^{n} |b_i|$. During one pass through the main loop of the algorithm either S remains unaltered (first case), or S decreases by at least one (second case). Since the value of k is increased by one in the first case, it follows that a particular value of S can occur for at most $(n+1)$ different values for k. But S can have at most $OD(b_1, b_2, \ldots, b_n) + 1$ different values, so that the number of passes through the main loop is $O(n (OD(b_1, b_2, \ldots, b_n) + 1))$.

The result now follows by observing that (1.12) takes $O(k^2)$ and that (1.13) takes $O(n k B)$ operations in $\mathbb{F}_q$. □

(1.15) Remark. With $OD(b_1, b_2, \ldots, b_n) \leq n B$ it follows that Algorithm (1.7) takes $O(n^4 B^2)$ arithmetic operations in $\mathbb{F}_q$.

(1.16) Remark. Most of the results above can be generalized to the case that L is a lattice in $\mathbb{F}_q[Y]^n$ of rank smaller than n. Let m be a

positive integer $< n$, let $b_1, b_2, \ldots, b_m \in \mathbb{F}_q[Y]^n$ be linearly independent over $\mathbb{F}_q[Y]$, and let L be the lattice in $\mathbb{F}_q[Y]^n$ of rank m spanned by $b_1, b_2, \ldots, b_m$:

$$L = \Sigma_{i=1}^{m} \mathbb{F}_q[Y] b_i .$$

By B we denote the $m \times n$ matrix having $b_1, b_2, \ldots, b_m$ as rows. We define the norm $|L|$ of L as the maximum of the norms of the determinants of the $m \times m$ submatrices of B; notice that $|L| = |d(L)|$ if $m = n$. This enables us to define the orthogonality defect $OD(b_1, b_2, \ldots, b_m)$ as $\Sigma_{i=1}^{m} |b_i| - |L|$. The basis $b_1, b_2, \ldots, b_m$ is reduced if the coordinates of $b_1, b_2, \ldots, b_m$ can be permuted in such a way that (1.8), (1.10), and (1.11) hold with k replaced by m. For $x \in L$ we denote by $\tilde{x} \in \mathbb{F}_q[Y]^m$ the vector consisting of the first m coordinates of x after application of the above permutation.

If the basis $b_1, b_2, \ldots, b_m$ is reduced, then $|b_j|$ is a j-th successive minimum of L. Namely, suppose that $|x|$ is a j-th successive minimum of L, for some $x \in L$. As in (1.2) we prove that $|\tilde{x}| \geq |\tilde{b}_j|$, so that, combined with $|x| \geq |\tilde{x}|$ and $|\tilde{b}_j| = |b_j|$, we find $|x| \geq |b_j|$.

It is easily verified (cf. (1.14)) that it takes $O(m^2 n (OD(b_1, b_2, \ldots, b_m) + 1) (\max_{1 \leq i \leq m} |b_i| + 1))$ operations in $\mathbb{F}_q$ to transform a basis $b_1, b_2, \ldots, b_m$ into a reduced one by means of Algorithm (1.7).

(1.17) Remark. We have given an algorithm to find successive minima in a lattice $L \subset \mathbb{F}_q[Y]^n$, and in particular the algorithm finds a shortest vector in L. In the sequel we will use this algorithm to decide whether L contains a non-zero element x satisfying $|x| \leq \ell$, for a certain small value of $\ell \geq 0$. This problem, however, can also be solved in a more direct way.

Suppose that a basis $b_1, b_2, \ldots, b_n$ for L is given, and that $OD(b_1, b_2, \ldots, b_n)$ is known. If an element x in L exists with $|x| \leq \ell$, then $x = \Sigma_{i=1}^{n} r_i b_i$ for certain polynomials $r_i \in \mathbb{F}_q[Y]$, with $|r_i| \leq \ell + OD(b_1, b_2, \ldots, b_n) - |b_i|$ (cf. (1.1)). Regarding the coefficients of r_i for $1 \leq i \leq n$ as unknowns, we can see this as a system of $n\,OD(b_1, b_2, \ldots, b_n)$ equations in $\Sigma_{i=1}^{n} (|r_i| + 1)$ unknowns over $\mathbb{F}_q$ (namely, for $1 \leq j \leq n$, the j-th coordinate of x equals $\Sigma_{i=1}^{n} r_i b_{ij} \in \mathbb{F}_q[Y]$, so that the $(\ell+1)$-th up to the $(\ell + OD(b_1, b_2, \ldots, b_n))$-th coefficient of $\Sigma_{i=1}^{n} r_i b_{ij}$ must be zero). Clearly, such an element x exists if and only if this system of equations over $\mathbb{F}_q$ has a solution. This results in an algorithm that takes $O(n^6 B^3)$ arithmetic operations in $\mathbb{F}_q$. An advantage of this method over Algorithm (1.7) is that, if we replace $\mathbb{F}_q$ by, for instance, the set of integers $\mathbb{Z}$, the coefficient growth during the Gaussian elimination can easily be bounded using methods from [2]. If we restrict ourselves to $\mathbb{F}_q$ however, then Algorithm (1.7) yields a better running time.

2. Factorization of polynomials in $\mathbb{F}_q[X, Y]$.

In this section we present an algorithm for the factorization of polynomials in two variables over a finite field that is polynomial-time in the degrees of the polynomial to be factored. The propositions and algorithms here are very similar to their counterparts in [7: Section 2, Section 3]. We therefore omit most of the details.

Let $f \in \mathbb{F}_q[X, Y]$ be the polynomial to be factored. Suppose that a positive integer u, and an irreducible polynomial $F \in \mathbb{F}_q[Y]$ of degree u are given. In the sequel we will describe how u and F are chosen. We may assume that F has leading coefficient one.

Let k be some positive integer. By (F^k) we denote the ideal generated by F^k. Since $\mathbb{F}_q[Y]/(F^k) \simeq \{\Sigma_{i=0}^{uk-1} a_i \alpha^i : a_i \in \mathbb{F}_q\}$, where $\alpha = (Y \bmod (F^k))$ is a zero of F^k, we can represent the elements of the ring $\mathbb{F}_q[Y]/(F^k)$ as polynomials in α over $\mathbb{F}_q$ of degree $< uk$. Notice that $\mathbb{F}_q[Y]/(F) \simeq \mathbb{F}_{q^u}$, the finite field containing q^u elements.

For a polynomial $g = \Sigma_i b_i X^i \in \mathbb{F}_q[X, Y]$, we denote by $(g \bmod F^k) \in (\mathbb{F}_q[Y]/(F^k))[X]$ the polynomial $\Sigma_i (b_i \bmod (F^k)) X^i$, and by $\delta_X g$ and $\delta_Y g$ the degrees of g in X and Y respectively.

Suppose that a polynomial $h \in \mathbb{F}_q[X, Y]$ is given such that:

(2.1) The leading coefficient with respect to X of h equals one,

(2.2) $(h \bmod F^k)$ divides $(f \bmod F^k)$ in $(\mathbb{F}_q[Y]/(F^k))[X]$,

(2.3) $(h \bmod F)$ is irreducible in $\mathbb{F}_{q^u}[X]$,

(2.4) $(h \bmod F)^2$ does not divide $(f \bmod F)$ in $\mathbb{F}_{q^u}[X]$.

Clearly $0 < \delta_X h \leq \delta_X f$. In the sequel we will see how such a polynomial h can be determined. The following proposition and its proof are similar to [7: (2.5)].

(2.5) Proposition. The polynomial f has an irreducible factor $h_0 \in \mathbb{F}_q[X, Y]$ for which $(h \bmod F)$ divides $(h_0 \bmod F)$ in $\mathbb{F}_{q^u}[X]$, and this factor is unique up to units in $\mathbb{F}_q$. Further, if g divides f in $\mathbb{F}_q[X, Y]$, then the following three assertions are equivalent:

(i) $(h \bmod F)$ divides $(g \bmod F)$ in $\mathbb{F}_{q^u}[X]$;

(ii) $(h \bmod F^k)$ divides $(g \bmod F^k)$ in $(\mathbb{F}_q[Y]/(F^k))[X]$;

(iii) h_0 divides g in $\mathbb{F}_q[X, Y]$.

In particular $(h \bmod F^k)$ divides $(h_0 \bmod F^k)$ in $(\mathbb{F}_q[Y]/(F^k))[X]$. □

(2.6) Let m be an integer $\geq \delta_X h$. Define L as the collection of polynomials $g \in \mathbb{F}_q[X, Y]$ with $\delta_X g \leq m$ and such that $(h \bmod F^k)$ divides $(g \bmod F^k)$ in $(\mathbb{F}_q[Y]/(F^k))[X]$. This is a subset of the $(m+1)$-dimensional vector space $\mathbb{F}_q(Y) + \mathbb{F}_q(Y)\,X + \ldots + \mathbb{F}_q(Y)\,X^m$. We identify this vector space with $\mathbb{F}_q(Y)^{m+1}$ by identifying $\Sigma_{i=0}^{m} a_i X^i \in \mathbb{F}_q(Y)[X]$ with $(a_0, a_1, \ldots, a_m)$. As in Section 1 the *norm* $|g|$ of the vector identified with the polynomial $g \in \mathbb{F}_q[X, Y]$ is defined as $\delta_Y g$. The collection L is a lattice in $\mathbb{F}_q[Y]^{m+1} \subset \mathbb{F}_q(Y)^{m+1}$ and, because of (2.1), a basis for L is given by

$$\{F^k X^i : 0 \leq i < \delta_X h\} \cup \{h X^{i - \delta_X h} : \delta_X h \leq i \leq m\}.$$

(2.7) Proposition. Let $b \in L$ satisfy

$$\delta_Y f\, \delta_X b + \delta_Y b\, \delta_X f < u k\, \delta_X h. \tag{2.8}$$

Then b is divisible by h_0 in $\mathbb{F}_q[X, Y]$, where h_0 is as in (2.5), and in particular $\gcd(f, b) \neq 1$.

Proof. We give only a sketch of the proof; for the details we refer to the proof of [7: (2.7)].

Put $g = \gcd(f, b)$, and $e = \delta_X g$. The projections of the polynomials

$$\{X^i f : 0 \leq i < \delta_X b - e\} \cup \{X^i b : 0 \leq i < \delta_X f - e\} \tag{2.9}$$

on $\mathbb{F}_q[Y]\,X^e + \mathbb{F}_q[Y]\,X^{e+1} + \ldots + \mathbb{F}_q[Y]\,X^{\delta_X f + \delta_X b - e - 1}$ form a basis for a $(\delta_X f + \delta_X b - 2e)$-dimensional lattice M' contained in $\mathbb{F}_q[Y]^{\delta_X f + \delta_X b - 2e}$. Define the *determinant* $d(M') \in \mathbb{F}_q[Y]$ of M' as the determinant of the matrix having these projections as rows, then we have

$$\delta_Y d(M') \leq \delta_Y f\,(\delta_X b - e) + \delta_Y b\,(\delta_X f - e).$$

Combined with (2.8) we get

(2.10) $\qquad \delta_Y d(M') < u\,k\,\delta_X h.$

Let $v \in \mathbb{F}_q[X, Y]$ be some linear combination over $\mathbb{F}_q[Y]$ of the polynomials in (2.9) such that $\delta_X v < e + \delta_X h$. Assuming that $(h \bmod F)$ does not divide $(g \bmod F)$ in $\mathbb{F}_{q^u}[X]$, it is not difficult to prove that

(2.11) $\qquad (v \bmod F^k) = 0.$

Now choose a basis $b_e, b_{e+1}, \ldots, b_{\delta_X f+\delta_X b-e-1}$ for M' such that $\delta_X b_i = i$ for $e \le i < \delta_X f + \delta_X b - e$ (which is clearly possible because $\mathbb{F}_q[Y]$ is euclidean). The degree with respect to Y of the leading coefficient with respect to X of the first $\delta_X h$ of these vectors b_i is, according to (2.11), at least $u\,k$. Since $d(M')$ equals the product of the leading coefficients, we find that

$$\delta_Y d(M') \ge u\,k\,\delta_X h,$$

which is a contradiction with (2.10). We conclude that $(h \bmod F)$ divides $(g \bmod F)$ in $\mathbb{F}_{q^u}[X]$, which, combined with Proposition (2.5), proves Proposition (2.7). □

(2.12) Proposition. Suppose that $b_1, b_2, \ldots, b_{m+1}$ is a reduced basis for L (see (1.3), (1.4), (1.5)), and that

(2.13) $\qquad \delta_Y f\,m + \delta_Y f\,\delta_X f < u\,k\,\delta_X h.$

Let h_0 be as in (2.5). Then the following three assertions are equivalent:

(i) $\quad \delta_X h_0 \le m;$

(ii) $\quad \delta_Y b_1 \le \delta_Y f;$

(iii) $b_1 = d\,h_0$ for some $d \in \mathbb{F}_q[X]$.

Proof. Use (1.6), (2.7), and $\delta_Y h_0 \le \delta_Y f$. □

Now that we have formulated the counterparts of [7: (2.5), (2.6), (2.7), (2.13)] in (2.5), (2.6), (2.7), and (2.12) respectively, we are ready to present the algorithm for factorization in $\mathbb{F}_q[X, Y]$.

We may assume that $f = \Sigma_i f_i X^i \in \mathbb{F}_q[X, Y]$ is *primitive*, i.e. $\delta_Y \gcd(f_0, f_1, \ldots, f_{\delta_X f}) = 0$ in $\mathbb{F}_q[Y]$, and that $\delta_X f > 0$ and $\delta_Y f > 0$. In the sequel we show that F of degree u can be chosen in such a way that

(2.14) $\qquad u = O(\delta_X f^{\varepsilon}\, \delta_Y f^{\varepsilon})$ for every $\varepsilon > 0$

(where the constant factor involved in the O does only depend on ε, and not on q).

First we sketch an algorithm to determine the factor of f that has a prescribed factor $(h \bmod F)$ in $\mathbb{F}_{q^u}[X]$ (cf. (2.5)); this is done in the proof of the following proposition.

(2.15) Proposition. Let $h \in \mathbb{F}_q[X, Y]$ be given such that (2.1), (2.3), (2.4), and (2.2) with k replaced by 1, are satisfied. The polynomial h_0, as defined in (2.5), can be found in $O(\delta_X h_0\, \delta_X f^5\, \delta_Y f^2)$ arithmetic operations in $\mathbb{F}_q$.

Proof. If $\delta_X h = \delta_X f$, then $h_0 = f$. Suppose that $\delta_X h < \delta_X f$. We take $k \in \mathbb{Z}_{>0}$ minimal such that (2.13) holds with m replaced by $\delta_X f - 1$:

(2.16) $\qquad u\,(k-1)\,\delta_X h \le \delta_Y f\,(2\,\delta_X f - 1) < u\,k\,\delta_X h.$

We modify h in such a way that (2.2) also holds for h and this value

of k. This can be done by means of a suitable version of Hensel's lemma as described for instance in [9: p79-81] (remark that Hensel's lemma can be applied because of (2.4)). It can easily be verified that the number of arithmetic operations in $\mathbb{F}_q$ needed for this modification of h is

$$O(u\,\delta_X f\,\delta_Y f + u^2\,\delta_X f^3 + k^2 u^2\,\delta_X h(\delta_X f - \delta_X h)),$$

where we use the fact that arithmetic operations in $\mathbb{F}_{q^u}$ can be done in $O(u^2)$ operations in $\mathbb{F}_q$. Combined with (2.14) and (2.16) this becomes

$$O(u^2\,\delta_X f^3 + \delta_X f^3\,\delta_Y f^2). \tag{2.17}$$

For each of the values of $m = \delta_X h, \delta_X h + 1, \ldots, \delta_X f - 1$ in succession we apply Algorithm (1.7) to the $(m+1)$-dimensional lattice L as defined in (2.6). But we stop as soon as for one of the values of m we succeed in determining h_0 using Proposition (2.12). If this does not occur for any m, then $\delta_X h_0 > \delta_X f - 1$, so $h_0 = f$.

The norms of the initial vectors in the bases of the lattices are bounded by $1 + \delta_Y f(2\,\delta_X f - 1)/\delta_X h$ (cf. (2.16)). If $b_1, b_2, \ldots, b_m$ is a reduced basis then $OD(b_1, b_2, \ldots, b_m, b_{m+1}) \le |b_{m+1}|$. Combining these observations with (1.14) and (1.15), we find that the total cost of the lattice reductions is

$$O(\delta_X h_0^4\,\delta_X f^2\,\delta_Y f^2 + \Sigma_{i=\delta_X h+1}^{\delta_X h_0}\ \delta_X h_0^3\,\delta_X f\,\delta_Y f\,|b_i|)$$

arithmetic operations in $\mathbb{F}_q$. This proves (2.15). □

(2.18) Theorem. Let f be a polynomial in $\mathbb{F}_q[X, Y]$. Then the factorization of f into irreducible factors in $\mathbb{F}_q[X, Y]$ can be determined in $O(\delta_X f^6\,\delta_Y f^2 + \delta_X f^3\,p\,m + \delta_Y f^3\,p\,m)$ arithmetic operations in $\mathbb{F}_q$, where $q = p^m$.

Proof. The factorization of the gcd of the coefficients of f with respect to X can be computed in $O(\delta_Y f^3 p\, m)$ arithmetic operations in $\mathbb{F}_q$ according to [3: Section 5]. Because the computation of this gcd also satisfies the estimates in (2.18), we may assume that f is primitive. We give an outline of the algorithm to factor f, and we analyze its running time.

First we calculate the resultant $R(f, f') \in \mathbb{F}_q[Y]$ of f and its derivative f' with respect to X, using the algorithm from [4]. This computation takes $O(\delta_X f^5 \delta_Y f^2)$ arithmetic operations in $\mathbb{F}_q$. We assume that $R(f, f') \neq 0$; it is well-known how to deal with the case $R(f, f') = 0$ (cf. [7: (3.5)]). Notice that, if both $\frac{\partial f}{\partial X}$ and $\frac{\partial f}{\partial Y}$ are zero, then $f(X, Y) = g(X^p, Y^p) = (h(X, Y))^p$, for polynomials g, h in $\mathbb{F}_q[X, Y]$.

Next we determine a positive integer u and an irreducible polynomial $F \in \mathbb{F}_q[Y]$ of degree u in such a way that $R(f, f') \not\equiv 0$ modulo F. This can be done as follows. If $q > \delta_Y R = \delta_Y R(f, f')$, then we choose an element $s \in \mathbb{F}_q$ such that $(Y - s)$ does not divide $R(f, f')$, and we put $F = Y - s$ and $u = 1$. This can be done in $O(\delta_Y R^2)$ operations in $\mathbb{F}_q$; if we use the parallel evaluation scheme as described in [1: Corollary 2, p294] this can be improved to $O(\delta_Y R^{1+\varepsilon})$ for every $\varepsilon > 0$.

Otherwise, if $q \leq \delta_Y R$, we take $\bar{u} \in \mathbb{Z}_{>0}$ minimal such that $q^{\bar{u}} > \delta_Y R$, so $q^{\bar{u}-1} = O(\delta_Y R)$. We determine an irreducible polynomial $G \in \mathbb{F}_q[Y]$ of degree $\bar{u}$ with leading coefficient one. Since we can restrict ourselves during this search for G to polynomials having 0 or 1 as coefficient for $Y^{\bar{u}-1}$, and because an irreducibility test for a polynomial of degree $\bar{u}$ in $\mathbb{F}_q[Y]$ takes $O(\bar{u}^2 \log q + \bar{u}^3)$ operations in $\mathbb{F}_q$, the determination of G can be done in $O(q^{\bar{u}-1}(\bar{u}^2 \log q + \bar{u}^3))$, that is $O(\delta_Y R^{1+\varepsilon})$ operations

in $\mathbb{F}_q$. (Namely, G of degree $\bar{u}$ without multiple factors is irreducible if and only if the $\bar{u}\times\bar{u}$ matrix with $(X^{iq} - X^i)$ modulo G for $0 \le i < \bar{u}$ as columns, has co-rank one.) We put $\mathbb{F}_{q^{\bar{u}}} = \mathbb{F}_q[Y]/(G)$. Since $q^{\bar{u}} > \delta_Y R$, there is an element $\beta \in \mathbb{F}_{q^{\bar{u}}}$ such that $R(f, f') \not\equiv 0$ modulo $(Y - \beta)$. Such an element β can be found in $O(\delta_Y R^{1+\varepsilon_1})$ operations in $\mathbb{F}_{q^{\bar{u}}}$ by evaluating $R(f, f')$ in $\delta_Y R + 1$ distinct points of $\mathbb{F}_{q^{\bar{u}}}$ by means of the parallel evaluation scheme from [1]. Arithmetic operations in $\mathbb{F}_{q^{\bar{u}}}$ take $O(\bar{u}^2) = O(\delta_Y R^{\varepsilon_2})$ arithmetic operations in $\mathbb{F}_q$, so the determination of β can be done in $O(\delta_Y R^{1+\varepsilon})$ operations in $\mathbb{F}_q$, for every $\varepsilon > 0$. Finally, we compute $F \in \mathbb{F}_q[Y]$ of degree $u \le \bar{u}$ as the minimal polynomial of β, by looking for a linear dependence relation among $\beta^0, \beta^1, \ldots, \beta^{\bar{u}}$; this takes $O(\bar{u}^2 u)$ operations in $\mathbb{F}_q$. Clearly, F satisfies $R(f, f')$ modulo $F \not\equiv 0$.

We conclude that in both cases F and u can be found in $O(\delta_Y R^{1+\varepsilon})$ arithmetic operations in $\mathbb{F}_q$, for every $\varepsilon > 0$. Since $\delta_Y R \le \delta_Y f\,(2\,\delta_X f - 1)$ this satisfies the estimates in (2.18). Notice that (2.14) is satisfied.

We now apply Berlekamp's algorithm [3: Section 5] to compute the irreducible factorization of $(f \bmod F)$ in $\mathbb{F}_{q^u}[X]$. We may assume that the factors have leading coefficient one. This computation takes $O(\delta_X f^3 p\,m\,u)$ arithmetic operations in $\mathbb{F}_q$. This becomes $O(\delta_X f^{4+\varepsilon}\,\delta_Y f^{1+\varepsilon})$ if $u \ne 1$, because this only occurs in the case that $p^m \le \delta_Y R(f, f')$, so that $p\,m\,u = O(\delta_X f^{1+\varepsilon}\,\delta_Y f^{1+\varepsilon})$. Since (2.4) is satisfied for all irreducible factors $(h \bmod F)$ of $(f \bmod F)$ in $\mathbb{F}_{q^u}[X]$, due to the choice of F and u, the complete factorization of f can be found by repeated application of Proposition (2.15). This takes $O(\delta_X f^6\,\delta_Y f^2)$ operations in $\mathbb{F}_q$. This proves (2.18). □

3. Factorization of polynomials in $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$.

In this section we describe an algorithm to factor polynomials in more than two variables with coefficients in a finite field. The algorithm that we will present here makes use of the algorithm from the previous section. At the end of this section we briefly explain an alternative version of our algorithm that does not depend on the algorithm from Section 2.

Let $f \in \mathbb{F}_q[X_1, X_2, \ldots, X_t]$ be the multivariate polynomial to be factored, with the number of variables $t \geq 3$. By $\delta_i f = n_i$ we denote the degree of f in X_i; for simplicity we often use n instead of n_1. We may assume that $n_i \leq n_j$ for $1 \leq i < j \leq t$, and that $n_1 \geq 2$. We put $N_j = \prod_{i=j}^{t} (n_i + 1)$. We say that f is *primitive* if the gcd of the coefficients of f with respect to X_1 equals one (i.e. is a unit in $\mathbb{F}_q$).

Let $k_3, k_4, \ldots, k_t$ be a $(t-2)$-tuple of integers. For $g \in \mathbb{F}_q[X_1, X_2, \ldots, X_t]$ we denote by $\tilde{g}_j \in \mathbb{F}_q[X_1, X_2, X_{j+1}, X_{j+2}, \ldots, X_t]$ the polynomial

$$g \text{ modulo } ((X_3 - X_2^{k_3}), (X_4 - X_2^{k_4}), \ldots, (X_j - X_2^{k_j})),$$

for $2 \leq j \leq t$; i.e. $\tilde{g}_j$ is g with $X_2^{k_i}$ substituted for X_i, for $3 \leq i \leq j$. Notice that $\tilde{g}_2 = g$. We put $\tilde{g} = \tilde{g}_t$.

Suppose that an irreducible factor $\tilde{h} \in \mathbb{F}_q[X_1, X_2]$ of $\tilde{f}$ is given such that

(3.1) $\quad \tilde{h}^2$ does not divide $\tilde{f}$ in $\mathbb{F}_q[X_1, X_2]$ and $\delta_1 \tilde{h} > 0$.

As in (2.5) we define h_0 as the irreducible factor of f in $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$ for which $\tilde{h}$ divides $\tilde{h}_0$ in $\mathbb{F}_q[X_1, X_2]$; the polynomial h_0 is unique up to units in $\mathbb{F}_q$.

(3.2) Let m be an integer with $\delta_1\tilde{h} \le m < n$. We define L as the collection of polynomials g in $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$ such that:

(i) $\delta_1 g \le m$ and $\delta_i g \le n_i$ for $3 \le i \le t$,

(ii) $\tilde{h}$ divides $\tilde{g}$ in $\mathbb{F}_q[X_1, X_2]$.

This is a subset of the $(m+1)N_3$-dimensional vector space $\mathbb{F}_q(X_2) + \mathbb{F}_q(X_2)X_t + \ldots + \mathbb{F}_q(X_2)X_1^m X_3^{n_3} \ldots X_t^{n_t}$. We put $M = (m+1)N_3$. We identify this vector space with $\mathbb{F}_q(X_2)^M$ by identifying $\Sigma_{i=0}^{m}\Sigma_{j=0}^{n_3}\ldots\Sigma_{k=0}^{n_t} a_{ij\ldots k} X_1^i X_3^j \ldots X_t^k \in \mathbb{F}_q(X_2)[X_1, X_3, \ldots, X_t]$ with $(a_{00\ldots0}, a_{00\ldots1}, \ldots, a_{mn_3\ldots n_t})$. As in Section 1 the *norm* $|g|$ of the vector associated with the polynomial $g \in \mathbb{F}_q[X_1, X_2, \ldots, X_t]$ is defined as $\delta_2 g$. The collection L is a lattice in $\mathbb{F}_q[X_2]^M \subset \mathbb{F}_q(X_2)^M$ of rank $M - \delta_1\tilde{h}$ (cf. (1.16)), and a basis for L over $\mathbb{F}_q[X_2]$ is given by

$$\{X_1^i \Pi_{j=3}^{t} (X_j - X_2^{k_j})^{i_j} : 0 \le i \le m,\ 0 \le i_j \le n_j \text{ for } 3 \le j \le t, \text{ and } (i_3, i_4, \ldots, i_t) \ne (0, 0, \ldots, 0)\}$$

$$\cup \{\tilde{h} X_1^{i-\delta_1\tilde{h}} : \delta_1\tilde{h} \le i \le m\}.$$

(3.3) Proposition. Suppose that f does not contain multiple factors. If

$$k_j > \Sigma_{i=2}^{j-1} k_i (2n n_i - n_i) \tag{3.4}$$

for $3 \le j \le t$, where $k_2 = 1$, and if b is a non-zero element of L with $|b| \le n_2$, then h_0 divides b in $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$, and in particular $\gcd(f, b) \ne 1$.

Proof. First we prove that $\gcd(f, b) \ne 1$. Suppose that $\gcd(f, b) = 1$. This implies that the resultant $R = R(f, b) \in \mathbb{F}_q[X_2, X_3, \ldots, X_t]$ of f and b (with respect to the variable X_1) is unequal to zero. Since $\tilde{h}$ divides

both $\tilde{f}$ and $\tilde{b}$ ((3.2)(ii)), and because $\tilde{R} = R(\tilde{f}, \tilde{b})$, we also have $\tilde{R} = 0$. This implies that there is an index j with $3 \le j \le t$ such that

$$\tilde{R}_j = 0. \tag{3.5}$$

Because of (3.2)(i) and $|b| \le n_2$, we have that $\delta_j b \le n_j$ for $2 \le j \le t$. Therefore $\delta_j R \le m n_j + n n_j \le 2 n n_j - n_j$, and also $\delta_j \tilde{R}_{j-1} \le 2 n n_j - n_j$, for $3 \le j \le t$. Because $\tilde{R}_j = \tilde{R}_{j-1} \bmod (X_j - X_2^{k_j})$ we get $\delta_2 \tilde{R}_j \le \delta_2 \tilde{R}_{j-1} + k_j \delta_j \tilde{R}_{j-1} \le \delta_2 \tilde{R}_{j-1} + k_j (2 n n_j - n_j)$, so that, with $k_2 = 1$ and $\tilde{R}_2 = R$,

$$\delta_2 \tilde{R}_j \le \Sigma_{i=2}^{j} k_i (2 n n_i - n_i) \tag{3.6}$$

for $2 \le j \le t$. According to (3.5) there must be an index j with $3 \le j \le t$ such that $(X_j - X_2^{k_j})$ divides $\tilde{R}_{j-1}$, which implies that

$$k_j \le \delta_2 \tilde{R}_{j-1}.$$

Combined with (3.4) and (3.6) this is a contradiction, so that $\gcd(f, b) \ne 1$.

Suppose that h_0 does not divide b in $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$. Then h_0 does not divide $r = \gcd(f, b)$, so that $\tilde{h}$ divides $\tilde{f}/\tilde{r}$ in $\mathbb{F}_q[X_1, X_2]$. Because $\delta_i(\tilde{f}/\tilde{r}) \le n_i$ for $1 \le i \le t$, the same reasoning as above yields that $\gcd(f/r, b) \ne 1$. This is a contradiction with $r = \gcd(f, b)$ because f does not contain multiple factors. □

(3.7) Suppose that f does not contain multiple factors and that f is primitive. Let

$$k_j = \Pi_{i=2}^{j-1} (2 n n_i - 1) \tag{3.8}$$

for $3 \le j \le t$, and let $\tilde{h}$ be chosen such that (3.1) is satisfied. Notice that (3.8) implies that (3.4) holds. The divisor h_0 of f can be determined in the following way.

For each of the values of $m = \delta_1\tilde{h}, \delta_1\tilde{h}+1, \ldots, n-1$ in succession we apply Algorithm (1.7) to the lattice L as defined in (3.2) (cf. (1.16)). But we stop as soon as for one of the values of m we succeed in finding a vector b_1 in L with $|b_1| \leq n_2$ (cf. (1.6)). Then $b_1 = c\,h_0$ for some $c \in \mathbb{F}_q[X_3, X_4, \ldots, X_t]$ (cf. (3.3)), which enables us to compute h_0. (Notice that we can even get $c \in \mathbb{F}_q$ if we increase the rank of L by one at each step.)

If we didn't find a short enough vector in any of the lattices, then $\delta_1 h_0 > n-1$, so that $h_0 = f$.

(3.9) Proposition. Assume that the conditions in (3.7) are satisfied. The polynomial h_0 can be computed in $O(\delta_1 h_0\, 2^{2t-4}\, n^{2t-1}\, N_2^2\, N_3^4)$ arithmetic operations in $\mathbb{F}_q$.

Proof. We derive an upper bound B for the norm of the vectors in the initial basis for L. From (3.8) we have

$$\delta_2\tilde{f} \leq \Sigma_{j=2}^{t} n_j \Pi_{i=2}^{j-1} (2\,n\,n_i - 1)$$

so that

$$\delta_2\tilde{f} \leq (2n)^{t-2} \Pi_{i=2}^{t} n_i. \tag{3.10}$$

Because $\tilde{h}$ divides $\tilde{f}$ in $\mathbb{F}_q[X_1, X_2]$, this bound also holds for $\delta_2\tilde{h}$. With (3.2) it follows that

$$B = O((2n)^{t-2} N_2).$$

From (1.16) we now find that the applications of Algorithm (1.7) together can be done in $O((\delta_1 h_0 N_3)^4 B^2 + \Sigma_{i=\delta_1\tilde{h}+1}^{\delta_1 h_0} (\delta_1 h_0 N_3)^3 B(N_3 B))$ arithmetic operations in $\mathbb{F}_q$.

The final gcd computations in $\mathbb{F}_q[X_3, X_4, \ldots, X_t]$ can be performed in $O(\delta_1 h_0 n_2 N_3^5)$ operations in $\mathbb{F}_q$, according to [4]. □

(3.11) We describe an algorithm to compute the irreducible factorization of a primitive polynomial f in $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$.

We assume that f does not contain multiple factors. This implies that the resultant $R = R(f, f') \in \mathbb{F}_q[X_2, X_3, \ldots, X_t]$ of f and its derivative f' with respect to X_1 is unequal to zero. We take $k_3, k_4, \ldots, k_t$ as in (3.8). It follows from the reasoning in the proof of (3.3) that $\tilde{R} \neq 0$ for this choice of $k_3, k_4, \ldots, k_t$, so that $\tilde{f}$ does not contain multiple factors. By means of the algorithm from Section 2 we compute the irreducible factors $\tilde{h}$ of $\tilde{f}$ of degree > 0 in X_1. Because (3.1) holds for all factors $\tilde{h}$ of $\tilde{f}$ thus found, we can compute the irreducible factors of f by repeated application of the algorithm described in (3.7).

It is well-known how to deal with the case that f contains multiple factors; notice that special attention has to be paid to the case that $\frac{\partial f}{\partial X_i} = 0$ for $1 \leq i \leq t$.

(3.12) Theorem. Let f be a polynomial in $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$, with $\delta_i f = n_i$ and $n_i \leq n_j$ for $1 \leq i < j \leq t$. The factorization of f into irreducible factors in $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$ can be determined in $O((2n_1)^{2t} N_2^2 N_3^4 + (2n_1)^{3t-6} N_2^3 p m)$ arithmetic operations in $\mathbb{F}_q$, where $q = p^m$, and $N_j = \prod_{i=j}^{t} (n_i + 1)$.

Proof. First assume that f is primitive. We apply (3.11). From (3.10) and (2.18) it follows that the factors of f of degree > 0 in X_1 can be found in $O(n_1^6 (2n_1)^{2t-4} N_2^2 + (2n_1)^{3t-6} N_2^3 p m)$ operations in $\mathbb{F}_q$. Repeated

application of (3.7) takes $O((2n_1)^{2t} N_2^2 N_3^4)$ operations in $\mathbb{F}_q$ according to (3.9). If f contains multiple factors, the gcd g of f and f' can be computed in $O(n_1^{3t-1} N_2^2)$ operations in $\mathbb{F}_q$ (cf. [4]), and the same estimates as above are valid for the factorization of f/g because $\delta_i(f/g) \leq \delta_i f$. It follows that a primitive polynomial can be factored in $O((2n_1)^{2t} N_2^2 N_3^4 + (2n_1)^{3t-6} N_2^3 p\, m)$ arithmetic operations in $\mathbb{F}_q$.

Now consider the case that f is not primitive. The computation of the gcd $\mathrm{cont}(f)$ of the coefficients in $\mathbb{F}_q[X_2, X_3, \ldots, X_t]$ of f takes $O(n_1 n_2^{3t-4} N_3^2)$ operations in $\mathbb{F}_q$. Because $\delta_i f = \delta_i(\mathrm{cont}(f)) + \delta_i(f/\mathrm{cont}(f))$, the proof follows by repeated application of the above reasoning. □

(3.13) Remark. It is possible to replace the factor $\tilde{h}$ of $\tilde{f}$ in the above algorithm by a factor $(\tilde{h} \bmod F^k)$ of $(\tilde{f} \bmod F^k)$, for a suitably chosen irreducible polynomial $F \in \mathbb{F}_q[X_2]$ and a positive integer k. The presentation of the resulting algorithm becomes somewhat more complicated in that case, but the ideas remain basically the same. An advantage of the alternative formulation is that the algorithm doesn't depend on Theorem (2.18), and that the algorithm can be regarded as a direct generalization of the algorithm from Section 2.

Acknowledgements.

Suggestions by H.W. Lenstra, Jr. and R.H. Mak have led to considerable improvements of the algorithms in sections 1 and 2.

References.

1. A.V. Aho, J.E. Hopcroft, J.D. Ullman, The design and analysis of computer algorithms, Addison Wesley, Reading 1974.

2. E.H. Bareiss, Sylvester's identity and multistep integer-preserving Gaussian-elimination, Math. Comp. 22 (1968), 565-578.

3. E.R. Berlekamp, Factoring polynomials over large finite fields, Math. Comp. 24 (1970), 713-735.

4. W.S. Brown, The subresultant PRS algorithm, ACM Transactions on mathematical software 4 (1978), 237-249.

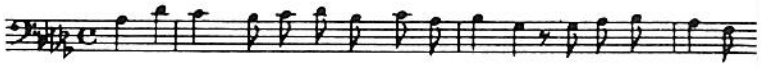
5. A.L. Chistov, D.Yu. Grigoryev, Polynomial-time factoring of multivariable polynomials over a global field, Lomi preprints E-5-82, Leningrad 1982.

6. E. Kaltofen, J. von zur Gathen, A polynomial-time factorization algorithm for multivariate polynomials over finite fields, manuscript, 1982.

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

8. K. Mahler, An analogue to Minkowski's geometry of numbers in a field of series, Annals of Mathematics 42 (1941), 488-522.

9. D.Y.Y. Yun, The Hensel lemma in algebraic manipulation, MIT, Cambridge 1974; reprint: Garland Publ. Co., New York 1980.



Factoring multivariate integral polynomials

A.K. Lenstra
Mathematisch Centrum
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

Abstract.

We present an algorithm to factor polynomials in several variables with integral coefficients that is polynomial-time in the degrees of the polynomial to be factored. Our algorithm generalizes the algorithm presented in [7] to factor integral polynomials in one variable.

1. Introduction.

The problem of factoring polynomials with integral coefficients remained open for a long time, i.e. no polynomial-time factoring algorithm was known. The best known algorithms took exponential-time in the worst case; these algorithms had to consider a possibly exponential number of combinations of p-adic factors before the true factors could be found or irreducibility could be decided. In [1] it was proven that the problem of factorization in $\mathbb{Z}[X]$ belongs to NP $\cap$ co-NP, which made its membership of P quite likely [2]. That this was indeed the case, was proven in [7] where a polynomial-time algorithm for factoring in $\mathbb{Z}[X]$ was given. This algorithm is based on the following three observations:

(1.1) The multiples of degree $< m$ of a p-adic factor together form a lattice in $\mathbb{Z}^m$;

(1.2) If this p-adic factor is computed up to a high enough precision, then the factor we are looking for is the shortest vector in this lattice;

(1.3) An approximation of the shortest vector in such a lattice can be found in polynomial-time by means of the so-called *basis reduction algorithm*.

In this paper we show that (1.1) and (1.2) can be generalized to polynomials in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ in an elementary way, for any $t \geq 2$. Combined with the same basis reduction algorithm as in (1.3), this leads to a polynomial-time algorithm for factoring in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$. In [8, 9, 10] we show that the above three points can be applied to various other kinds of polynomial factoring problems as well (like multi-

variate polynomials over finite fields or over algebraic number fields). Another approach to multivariate integral polynomial factorization is given in [5]. There the multivariate case is first reduced in polynomial-time to the bivariate case, next bivariate is reduced to univariate.

For practical purposes we do not recommend any of these polynomial-time algorithms; their running time will be dominated by the rather slow basis reduction algorithm. For polynomials in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ the algorithm from [12] for instance is very useful, although it is exponential-time in the worst case.

We restrict ourselves in this paper to integral polynomials in two variables; the multivariate case follows immediately from this. In Section 2 we present an important result from [7: Section 1] concerning the basis reduction algorithm mentioned in (1.3). The generalizations of (1.1) and (1.2) to polynomials in $\mathbb{Z}[X,Y]$ are described in Section 3, and in Section 4 we give an outline of the factoring algorithm, and we analyze its running time.

2. The basis reduction algorithm.

The basis reduction algorithm from [7: Section 1] makes it possible to determine in polynomial-time a reasonable approximation of the shortest vector in a lattice. We will not give a description of the algorithm here. It will suffice to summarize those results from [7: Section 1] that we will need here.

Let $b_1, b_2, \ldots, b_n \in \mathbb{Z}^n$ be linearly independent. For our purposes we may assume that the $n \times n$ matrix having $b_1, b_2, \ldots, b_n$ as columns is upper-triangular. The i-dimensional lattice $L_i \subset \mathbb{Z}^i$ with basis $b_1, b_2, \ldots, b_i$ is defined as $L_i = \Sigma_{j=1}^{i} \mathbb{Z}\, b_j = \{\Sigma_{j=1}^{i} r_j b_j : r_j \in \mathbb{Z}\}$. We put $L = L_n$.

(2.1) Proposition. (cf. [7: (1.11), (1.26), (1.37)]) Let $B \in \mathbb{Z}_{\geq 2}$ be such that $|b_j|^2 \leq B$ for $1 \leq j \leq n$, where $|\ |$ denotes the ordinary Euclidean length. The basis reduction algorithm as described in [7: (1.15)] determines a vector $\tilde{b} \in L$ such that $\tilde{b}$ belongs to a basis for L, and such that $|\tilde{b}|^2 \leq 2^{n-1}|x|^2$ for every $x \in L$, $x \neq 0$; the algorithm takes $O(n^4 \log B)$ elementary operations on integers having binary length $O(n \log B)$. Furthermore, during the first $O(i^4 \log B)$ operations (on integers having binary length $O(i \log B)$), vectors $\tilde{b}_i \in L_i$, belonging to a basis for L_i, are deter-

mined such that $|\tilde{b}_i|^2 \le 2^{i-1}|x_i|^2$ for every $x_i \in L_i$, $x_i \neq 0$, for $1 \le i \le n$. □

So, we can find a reasonable approximation of the shortest vector in L in polynomial-time. But also we find, during this computation, approximations of the shortest vectors of the lattices L_i without any time loss.

3. Factors and lattices.

We describe how to generalize (1.1) and (1.2) to polynomials in $\mathbb{Z}[X,Y]$. Let $f \in \mathbb{Z}[X,Y]$ be the polynomial to be factored; we may assume that f has no multiple factors, i.e. f is *square-free*. Furthermore we assume that f is *primitive* with respect to X, i.e. the greatest common divisor of the coefficients in $\mathbb{Z}[Y]$ of f equals one. We denote by $\delta_X f$ and $\delta_Y f$ the degrees of f in X and Y respectively, and by $\ell c(f)$ the *leading coefficient* of f with respect to X. We put $n_X = \delta_X f$ and $n_Y = \delta_Y f$.

Suppose that we are given a prime number p, an integer s and a positive integer k. By (s_1) we denote the ideal generated by p and $(Y-s)$, and by (s_k) we denote the ideal generated by p^k and $(Y-s)^{n_Y+1}$. In Section 4 we will see how to find a polynomial $h \in \mathbb{Z}[X,Y]$ such that:

(3.1) $\ell c(h) = 1$,

(3.2) $(h \bmod (s_k))$ divides $(f \bmod (s_k))$ in $\mathbb{Z}[X,Y]/(s_k)$,

(3.3) $(h \bmod (s_1)) \in (\mathbb{Z}/p\mathbb{Z})[X]$ is irreducible in $(\mathbb{Z}/p\mathbb{Z})[X]$,

(3.4) $(h \bmod (s_1))^2$ does not divide $(f \bmod (s_1))$ in $(\mathbb{Z}/p\mathbb{Z})[X]$.

We put $\ell = \delta_X h$; so $0 < \ell \le n_X$.

Let $h_0 \in \mathbb{Z}[X,Y]$ be the irreducible factor of f for which $(h \bmod (s_1))$ divides $(h_0 \bmod (s_1))$ in $(\mathbb{Z}/p\mathbb{Z})[X]$ (or equivalently $(h \bmod (s_k))$ divides $(h_0 \bmod (s_k))$ in $\mathbb{Z}[X,Y]/(s_k)$, cf. [7: (2.5)]); notice that h_0 is unique up to sign.

(3.5) Let m_X and m_Y be two integers with $\ell \le m_X < n_X$ and $0 \le m_Y \le \delta_Y \ell c(f)$. We define L as the collection of polynomials $g \in \mathbb{Z}[X,Y]$ such that

(i) $\delta_X g \le m_X$,

(ii) $\delta_Y g \le n_Y$,

(iii) $\delta_Y \ell c(g) \le m_Y$,

(iv) $(h \bmod (s_k))$ divides $(g \bmod (s_k))$ in $\mathbb{Z}[X,Y]/(s_k)$.

Putting $M = m_X(n_Y+1)+m_Y+1$ it is not difficult to see that L is an M-dimensional lattice contained in $\mathbb{Z}^M$, where we identify polynomials in L and M-dimensional vectors in the usual way (i.e. $\sum_{i=0}^{m_X-1}\sum_{j=0}^{n_Y} a_{ij}X^iY^j + \sum_{j=0}^{m_Y} a_{m_X j}X^{m_X}Y^j$ is identified with $(a_{00}, a_{01}, \ldots, a_{0n_Y}, a_{10}, \ldots, a_{m_X-1\,n_Y}, a_{m_X 0}, \ldots, a_{m_X m_Y})$). Because of (3.1) a basis for L is given by

$$\{p^{k_Y j}X^i:\ 0 \le j \le n_Y,\ 0 \le i < \ell\} \cup$$
$$\{(hY^j \bmod (s_k))X^{i-\ell}:\ (0 \le j \le n_Y \text{ and } \ell \le i < m_X) \text{ or } (0 \le j \le m_Y \text{ and } i = m_X)\}.$$

This generalizes (1.1) (cf. [7: (2.6)]). We now come to (1.2). The *height* g_{max} of a polynomial g is defined as the maximal absolute value of any of its integral coefficients. We prove that, if k and s are suitably chosen, then a vector of small height in L must lead to a factorization of f.

(3.6) Proposition. Suppose that $g \in L$ satisfies

$$|s|^{n_Y+1} > (e^{n_X+n_Y} f_{max}\sqrt{(n_X+1)(n_Y+1)})^{m_X}(g_{max}\sqrt{(m_X+1)(n_Y+1)})^{n_X} \tag{3.7}$$

and

$$p^k > (e^{n_X+n_Y} f_{max}\sqrt{(n_X+1)(n_Y+1)})^{m_X}(g_{max}\sqrt{(m_X+1)(n_Y+1)})^{n_X}(1+(1+|s|)^{n_Y+1})^{n_Y(n_X+m_X-1)}. \tag{3.8}$$

Then h_0 divides g in $\mathbb{Z}[X,Y]$, and in particular $\gcd(f,g) \neq 1$.

Proof. Suppose that $\gcd(f,g) = 1$. This implies that the resultant $R \in \mathbb{Z}[Y]$ of f and g is unequal to zero. Using the result from [4] one proves that

$$|R| < (f_{max}\sqrt{(n_X+1)(n_Y+1)})^{m_X}(g_{max}\sqrt{(m_X+1)(n_Y+1)})^{n_X}, \tag{3.9}$$

where $|R|$ denotes the ordinary Euclidean length of the vector identified with R. Since $(h \bmod (s_k))$ divides both $(f \bmod (s_k))$ and $(g \bmod (s_k))$, the polynomials f and g have a non-trivial common divisor in $\mathbb{Z}[X,Y]/(s_k)$, so that R must be zero modulo the ideal generated by p^k and $(Y-s)^{n_Y+1}$. The polynomial $(Y-s)^{n_Y+1}$ cannot divide R, because this would imply, according to [11: Theorem 1], that $|s|^{n_Y+1} \le |R|$, which is, combined with (3.9), a contradiction with (3.7). Therefore $(R \bmod (Y-s)^{n_Y+1})$ has to be zero modulo p^k. Using induction on n_Y+1 it is easy to prove that

$$(R \bmod (Y-s)^{n_Y+1})_{max} \le R_{max}(1+(1+|s|)^{n_Y+1})^{n_Y(n_X+m_X-1)},$$

so that, with $R_{max} \le |R|$ and (3.8), it follows that $(R \bmod (Y-s)^{n_Y+1})$ cannot be zero

modulo p^k. We conclude that $\gcd(f,g) \neq 1$.

Suppose that h_0 does not divide g. So h_0 does not divide $r = \gcd(f,g)$, so $(h \bmod (s_k))$ divides $((f/r) \bmod (s_k))$. Because f/r divides f, we find from [3] that $(f/r)_{max} \leq e^{n_X+n_Y} f_{max}$. This implies that the above reasoning applies to f/r and the same polynomial g in L, so that $\gcd(f/r,g) \neq 1$. This is a contradiction with $r = \gcd(f,g)$, because f is square-free. □

(3.10) Proposition. Suppose that s and k are chosen in such a way that (3.7) and (3.8) are satisfied with g_{max} replaced by $2^{(M-1)/2}\sqrt{M}\, e^{n_X+n_Y} f_{max}$. Let $\tilde{b}$ be as in (2.1) the result of an application of the basis reduction algorithm to the M-dimensional lattice L as defined in (3.5). Then $h_0 \in L$ if and only if (3.7) and (3.8) are satisfied with g replaced by $\tilde{b}$.

Proof. To prove the "if"-part, assume that (3.7) and (3.8) hold with g_{max} replaced by $\tilde{b}_{max}$. According to (3.6) this implies that h_0 divides $\tilde{b}$, so that $h_0 \in L$.

To prove the "only if"-part, assume that $h_0 \in L$. Because h_0 divides f, we find from [3] that $(h_0)_{max} \leq e^{n_X+n_Y} f_{max}$. So there exists a non-zero vector in L with Euclidean length bounded by $\sqrt{M}\, e^{n_X+n_Y} f_{max}$. Application of (2.1) yields that $\tilde{b}_{max} \leq |b| \leq 2^{(M-1)/2}\sqrt{M}\ e^{n_X+n_Y} f_{max}$. Combined with the above choices of s and k, this implies that (3.7) and (3.8) hold with g replaced by $\tilde{b}$. □

4. Description of the algorithm.

In this section we present the polynomial-time algorithm to factor f. First we give an algorithm to determine the factor h_0, given p, s and h. After that, we will see how p and s have to be chosen.

(4.1) Let p, s and h be as in Section 3, such that (3.1), (3.3), (3.4) and (3.2) with k replaced by 1 are satisfied. Assume that s satisfies the condition in (3.10) with m_X and m_Y replaced by n_X-1 and $\delta_Y \ell c(f)$ respectively:

$$(4.2) \qquad |s|^{n_Y+1} > \left(e^{n_X+n_Y} f_{max} \sqrt{(n_X+1)(n_Y+1)}\right)^{n_X-1} \left(2^{(M-1)/2}\sqrt{M}\, e^{n_X+n_Y} f_{max} \sqrt{n_X(n_Y+1)}\right)^{n_X}$$

where $M = (n_X-1)(n_Y+1)+\delta_Y \ell c(f)+1$. We describe an algorithm that determines h_0, the irreducible factor of f such that $(h \bmod (s_1))$ divides $(h_0 \bmod (s_1))$ in $(\mathbb{Z}/p\mathbb{Z})[X]$.

We may assume that $\ell = \delta_X h < n_X$. Take k minimal such that the condition from (3.10) is satisfied with m_X and m_Y replaced by n_X-1 and $\delta_Y \ell c(f)$ respectively:

$$(4.3) \qquad p^k > (e^{n_X+n_Y} f_{max} \sqrt{(n_X+1)(n_Y+1)})^{n_X-1} (2^{(M-1)/2} \sqrt{M}\, e^{n_X+n_Y} f_{max} \sqrt{n_X(n_Y+1)})^{n_X} \cdot (1+(1+|s|)^{n_Y+1})^{2n_Y(n_X-1)}.$$

Next modify h in such a way that (3.2) also holds for this value of k; because of (3.4) this can be done by means of Hensel's lemma [13].

Apply Proposition (2.1) to the M-dimensional lattice L as defined in (3.5) for each of the values of $M = \ell(n_Y+1)+1, \ell(n_Y+1)+2, \ldots, \ell(n_Y+1)+\delta_Y \ell c(f)+1, (\ell+1)(n_Y+1)+1, \ldots, (n_X-1)(n_Y+1)+\delta_Y \ell c(f)+1$ in succession (so, for $m_X = \ell, \ell+1, \ldots, n_X-1$ in succession and for every value of m_X the values $m_Y = 0, 1, \ldots, \delta_Y \ell c(f)$ in succession). But stop as soon as a vector $\tilde{b}$ is found satisfying (3.7) and (3.8) with g replaced by $\tilde{b}$.

If such a vector $\tilde{b}$ is found for a certain value of M ($m_X = m_{X0}$ and $m_Y = m_{Y0}$), then we know from (3.10) that $h_0 \in L$. Since we try the values of M in succession this implies that $\delta_X h_0 = m_{X0}$ and $\delta_Y \ell c(h_0) = m_{Y0}$. By (3.6) h_0 divides $\tilde{b}$, so that $\delta_X \tilde{b} = m_{X0}$ and $\delta_Y \ell c(\tilde{b}) = m_{Y0}$. So $\tilde{b} = ch_0$ for some $c \in \mathbb{Z}$, but $h_0 \in L$ and $\tilde{b}$ belongs to a basis for L, so $\tilde{b} = \pm h_0$.

If no such vector $\tilde{b}$ was found, then (3.10) implies that $\delta_X h_0 > n_X - 1$, so that $h_0 = f$, because f is primitive.

This finishes the description of Algorithm (4.1).

(4.4) Proposition. Denote by $m_{X0} = \delta_X h_0$ the degree in X of the irreducible factor h_0 of f that is found by Algorithm (4.1). Then the number of arithmetic operations needed by Algorithm (4.1) is $O(m_{X0}(n_X^5 n_Y^5 + n_X^4 n_Y^4 \log(f_{max}) + n_X^4 n_Y^6 \log(|s|) + n_X^3 n_Y^4 \log p))$ and the integers on which these operations have to be performed each have binary length $O(n_X^3 n_Y^2 + n_X^2 n_Y \log(f_{max}) + n_X^2 n_Y^3 \log(|s|) + n_X n_Y \log p)$.

Proof. Let M_1 be the largest value of M for which (2.1) is applied; so $M_1 = O(m_{X0} n_Y)$. It follows from (2.1) that the number of operations needed for the applications of the basis reduction algorithm for $\ell(n_Y+1)+1 \le M \le M_1$ is equal to the number of operations needed for $M = M_1$ only. Assuming that the coefficients of the initial basis for L are reduced modulo p^k, we find, using (4.3), that the following holds for the bound

B on the length of these vectors:

$$\log B = O(n_X^2 n_Y + n_X \log(f_{max}) + n_X n_Y^2 \log(|s|) + \log p).$$

With $M_1 = O(m_{X0} n_Y)$ and (2.1) this gives the estimates in (4.4).

The verification that the same estimates are valid for the application of Hensel's lemma is straightforward [13]. □

We now describe how s and p have to be chosen. First, s must be chosen such that $(f \bmod (Y-s)) = f(X,s)$ remains square-free, and such that (4.2) holds. The resultant R of f and its derivative f' with respect to X is a non-zero polynomial in $\mathbb{Z}[Y]$ of degree $\leq n_Y(2n_X-1)$. Therefore we can find in $O(n_X n_Y)$ trials the minimal integer s such that s is not a zero of R, and such that (4.2) holds. It is easily verified that $\log(|s|) = O(n_X^2 + n_X \log(f_{max}))$.

Next we choose p as the smallest prime number not dividing the resultant of f(X,s) and f'(X,s). Since $\log(f(X,s)_{max}) = O(n_X^2 n_Y + n_X n_Y \log(f_{max}))$, it follows as in the proof of [7: (3.6)] that $p = O(n_X^3 n_Y + n_X^2 n_Y \log(f_{max}))$.

The complete factorization of $(f \bmod (s_1))$ can be determined by means of Berlekamp's algorithm [6: section 4.6.2]; notice that (3.4) holds for every factor $(h \bmod (s_1))$ of $(f \bmod (s_1))$, because of the choice of p, and that this factorization can be found in polynomial-time, because of the bound on p. The algorithm to factor f completely now follows by repeated application of Algorithm (4.1). The above bounds on $\log(|s|)$ and p, combined with (4.4) and the fact that a factor g of f satisfies $\log(g_{max}) = O(n_X + n_Y + \log(f_{max}))$ (cf. [3]), yields the following theorem.

(4.5) Theorem. The number of arithmetic operations needed to factor f completely is $O(n_X^7 n_Y^6 + n_X^6 n_Y^6 \log(f_{max}))$, and the integers on which these operations have to be performed each have binary length $O(n_X^4 n_Y^3 + n_X^3 n_Y^3 \log(f_{max}))$. □

5. Conclusion.

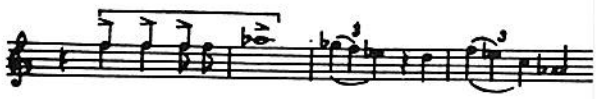
We have shown that basically the same ideas that were used for the polynomial-time algorithm for factoring in $\mathbb{Z}[X]$ lead to a polynomial-time factoring algorithm in $\mathbb{Z}[X,Y]$ (Theorem (4.5)). Our method can be generalized to polynomials in $\mathbb{Z}[X_1, X_2,$

$\ldots, X_t]$. The evaluation $(Y = s)$ is then replaced by $(X_2 = s_2, X_3 = s_3, \ldots, X_t = s_t)$, where the integers s_i have to satisfy conditions similar to (4.2). It will not be surprising that in this case the estimates become rather complicated.

A somewhat simpler algorithm results if we use the algorithm from [7]; the details of this algorithm, which is similar to the one described in this paper, can be found in [10].

References.

1. D.G. Cantor, Irreducible polynomials with integral coefficients have succinct certificates, J. of Algorithms 2 (1981), 385-392.
2. M.R. Garey, D.S. Johnson, Computers and intractability, Freeman, San Francisco 1979.
3. A.O. Gel'fond, Transcendental and algebraic numbers, Dover Publ., New York 1960.
4. A.J. Goldstein, R.L. Graham, A Hadamard-type bound on the coefficients of a determinant of polynomials, SIAM Rev. 16 (1974), 394-395.
5. E. Kaltofen, On the complexity of factoring polynomials with integer coefficients, Ph.D. thesis, Rensselaer Polytechnic Institute, August 1982.
6. D.E. Knuth, The art of computer programming, vol. 2, Seminumerical algorithms, Addison Wesley, Reading, second edition 1981.
7. A.K. Lenstra, H.W. Lenstra, Jr., L. Lovász, Factoring polynomials with rational coefficients, Math. Ann. 261 (1982), 515-534.
8. A.K. Lenstra, Factoring polynomials over algebraic number fields, Report IW 213/82, Mathematisch Centrum, Amsterdam 1982 (also Proceedings Eurocal '83).
9. A.K. Lenstra, Factoring multivariate polynomials over finite fields, Report IW 221/83, Mathematisch Centrum, Amsterdam 1983 (also Proceedings 15th STOC).
10. A.K. Lenstra, to appear.
11. M. Mignotte, An inequality about factors of polynomials, Math. Comp. 28 (1974), 1153-1157
12. P.S. Wang, An improved multivariate polynomial factoring algorithm, Math. Comp. 32 (1978), 1215-1231.
13. D.Y.Y. Yun, The Hensel lemma in algebraic manipulation, MIT, Cambridge 1974; reprint: Garland publ. Co., New York 1980.



Factoring multivariate integral polynomials, II

by

A.K. Lenstra

ABSTRACT

We show that the problem of factoring multivariate integral polynomials can be reduced in polynomial-time to the univariate case. Our reduction makes use of lattice techniques as introduced in [3].

KEY WORDS & PHRASES: *polynomial algorithm, polynomial factorization*

1. Introduction.

In [5] we presented a polynomial-time algorithm to factor polynomials in $\mathbb{Z}[X, Y]$, and we pointed out how to generalize the algorithm to $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ for $t \geq 3$. A nice feature of this algorithm is that it does not depend on the polynomial-time algorithm to factor in $\mathbb{Z}[X]$ (cf. [3]). Instead of working out the details of this direct approach for $t \geq 3$ (this will be done for $\mathbb{Q}(\alpha)[X_1, X_2, \ldots, X_t]$ in a forthcoming paper [6]), we here simplify the method from [5] somewhat, which results in a polynomial-time reduction from factoring in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ to factoring in $\mathbb{Z}[X]$. This reduction is similar to the reduction from $\mathbb{F}_q[X_1, X_2, \ldots, X_t]$ to $\mathbb{F}_q[X, Y]$ that was given in [4].

An outline of our reduction is as follows. First we evaluate the polynomial $f \in \mathbb{Z}[X_1, X_2, \ldots, X_t]$ in a suitably chosen integer point $(X_2 = s_2, X_3 = s_3, \ldots, X_t = s_t)$, to obtain a polynomial $\bar{f} \in \mathbb{Z}[X_1]$. Using the algorithm from [3] we then compute an irreducible factor $\bar{h} \in \mathbb{Z}[X_1]$ of $\bar{f}$. Next we construct an integral lattice containing the factor h_0 of f that corresponds to $\bar{h}$, and we prove that h_0 is the shortest vector in this lattice. As usual, this enables us to compute h_0 by means of the so-called *basis reduction algorithm* (cf. [3: Section 1]; in the sequel we will assume the reader to be familiar with this basis reduction algorithm and its properties).

2. Factoring multivariate integral polynomials.

Let $f \in \mathbb{Z}[X_1, X_2, \ldots, X_t]$ be the polynomial to be factored, with the number of variables $t \geq 2$. By $\delta_i f = n_i$ we denote the degree of f in X_i. We

often use n instead of n_1. We put $N_i = \prod_{k=i}^{t}(n_k+1)$, and $N = N_1$. The *content* $\mathrm{cont}(f) \in \mathbb{Z}[X_2, X_3, \ldots, X_t]$ of f is defined as the greatest common divisor of the coefficients of f with respect to X_1; we say that f is *primitive* if $\mathrm{cont}(f) = 1$.

Without loss of generality we may assume that $2 \le n_i \le n_{i+1}$ for $1 \le i < t$, and that the gcd of the integer coefficients of f equals one.

We present an algorithm to factor f into its irreducible factors in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ that is polynomial-time in N and the size of the integer coefficients of f.

Let $s_2, s_3, \ldots, s_t \in \mathbb{Z}_{>0}$ be a $(t-1)$-tuple of integers. For $g \in \mathbb{Z}[X_1, X_2, \ldots, X_t]$ we denote by $\tilde{g}_j$ the polynomial g modulo$((X_2-s_2), (X_3-s_3), \ldots, (X_j-s_j)) \in \mathbb{Z}[X_1, X_{j+1}, X_{j+2}, \ldots, X_t]$; i.e. $\tilde{g}_j$ is g with s_i substituted for X_i for $i = 2, 3, \ldots, j$. Notice that $\tilde{g}_1 = g$, and that $\tilde{g}_j = \tilde{g}_{j-1} \bmod (X_j-s_j)$. We put $\tilde{g} = \tilde{g}_t$.

Suppose that an irreducible, primitive factor $\tilde{h} \in \mathbb{Z}[X_1]$ of $\tilde{f}$ is given such that

(2.1) $\tilde{h}^2$ does not divide $\tilde{f}$ in $\mathbb{Z}[X_1]$, and $\delta_1\tilde{h} > 0$.

This condition implies that there exists an irreducible factor $h_0 \in \mathbb{Z}[X_1, X_2, \ldots, X_t]$ of f such that $\tilde{h}$ divides $\tilde{h}_0$ in $\mathbb{Z}[X_1]$, and that this polynomial h_0 is unique up to sign.

(2.2) Let m be an integer with $\delta_1\tilde{h} \le m < n$. We define L as the collection of polynomials g in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ such that

(i) $\delta_1 g \le m$, and $\delta_i g \le n_i$ for $2 \le i \le t$,

(ii) $\tilde{h}$ divides $\tilde{g}$ in $\mathbb{Z}[X_1]$.

This is a subset of the $(m+1)N_2$-dimensional real vector space $\mathbb{R} + \mathbb{R}X_t + \ldots +$

$\mathbb{R}X_1^m X_2^{n_2} \ldots X_t^{n_t}$. We put $M = (m+1)N_2$. This vector space can be identified with $\mathbb{R}^M$ by identifying the polynomial $\Sigma_{i=0}^{m} \Sigma_{j=0}^{n_2} \ldots \Sigma_{k=0}^{n_t} a_{ij\ldots k} X_1^i X_2^j \ldots X_t^k$ $\in \mathbb{R}[X_1, X_2, \ldots, X_t]$ with the M-dimensional vector $(a_{00\ldots0}, a_{00\ldots1}, \ldots, a_{mn_2\ldots n_t})$. The collection L is a lattice in $\mathbb{Z}^M$ of rank $M - \delta_1\tilde{n}$, and a basis for L is given by

$$\{X_1^i \Pi_{j=2}^{t} (X_j - s_j)^{i_j}: \ 0 \le i \le m, \ 0 \le i_j \le n_j \ \text{ for } \ 2 \le j \le t, \ \text{ and}$$

$$(i_2, i_3, \ldots, i_t) \neq (0, 0, \ldots, 0)\}$$

$$\cup \{\tilde{n} X_1^{i-\delta_1\tilde{n}}: \ \delta_1\tilde{n} \le i \le m\}$$

(cf. [4: (3.2)]).

We define the *length* $|g|$ of the vector associated with the polynomial g as the ordinary Euclidean length of this vector. The *height* g_{max} is defined as the largest absolute value of any of the integer coefficients of g.

(2.3) Proposition. *Suppose that* b *is a non-zero element of* L *such that*

$$(2.4) \qquad s_j \ge f_{max}^m \, b_{max}^n \, (n+m)! \, (N_2 \Pi_{i=2}^{j-1} s_i^{n_i})^{n+m}$$

for $2 \le j \le t$. *Then* $\gcd(f,b) \neq 1$ *in* $\mathbb{Z}[X_1, X_2, \ldots, X_t]$. *)

Proof. Suppose on the contrary that $\gcd(f,b) = 1$. This implies that the resultant $R = R(f,b) \in \mathbb{Z}[X_2, X_3, \ldots, X_t]$ of f and b (with respect to the variable X_1) is unequal to zero.

We derive an upper bound for $(\tilde{R}_j)_{max}$. Because $\tilde{R}_j$ is the resultant of $\tilde{f}_j$ and $\tilde{b}_j$ we have

$$(2.5) \qquad (\tilde{R}_j)_{max} \le (\tilde{f}_j)_{max}^m (\tilde{b}_j)_{max}^n (n+m)! \, N_{j+1}^{n+m-2}$$

*) Here, and in the sequel, f_{max}^m denotes $(f_{max})^m$.

as is easily verified. Because $\tilde{b}_j = \tilde{b}_{j-1} \text{ modulo } (X_j - s_j)$, we have

$$(\tilde{b}_j)_{max} \leq (\tilde{b}_{j-1})_{max} (n_j+1)\, s_j^{n_j},$$

so that

$$(2.6) \qquad (\tilde{b}_j)_{max} \leq b_{max} \prod_{i=2}^{j} (n_i+1) s_i^{n_i},$$

and similarly

$$(2.7) \qquad (\tilde{f}_j)_{max} \leq f_{max} \prod_{i=2}^{j} (n_i+1) s_i^{n_i}.$$

Combining (2.5), (2.6), and (2.7), we obtain

$$(2.8) \qquad (\tilde{R}_j)_{max} < f_{max}^{m}\, b_{max}^{n}\, (n+m)!\, \left(N_2 \prod_{i=2}^{j} s_i^{n_i}\right)^{n+m},$$

for $1 \leq j < t$.

Because $\tilde{n}$ divides both $\tilde{f}$ and $\tilde{b}$ ((2.2)(ii)), we have that $\tilde{R} = 0$. But also $R \neq 0$, so there must be an index j with $2 \leq j \leq t$ such that s_j is a zero of $\tilde{R}_{j-1}$. This implies that

$$|s_j| \leq (\tilde{R}_{j-1})_{max}$$

for some j with $2 \leq j \leq t$, which yields, combined with (2.4) and (2.8), a contradiction. We conclude that $\gcd(f,b) \neq 1$. □

(2.9) Proposition. *Let* $b_1, b_2, \ldots, b_M$ *be a reduced basis for* L *(cf.* [3: Section 1]*), where* L *and* M *are defined as in* (2.2). *Suppose that*

$$(2.10) \qquad s_j \geq f_{max}^{m} \left((M\, 2^{M-1})^{\frac{1}{2}} f_{max}\right)^{n} (n+m)! \left(e^{\sum_{i=1}^{t} n_i}\, N_2 \prod_{i=2}^{j-1} s_i^{n_i}\right)^{n+m}$$

for $2 \leq j \leq t$, *and that* f *does not contain multiple factors. Then*

(2.11) $\quad (b_1)_{max} \le (M\,2^{M-1})^{\frac{1}{2}}\, e^{\sum_{i=1}^{t} n_i}\, f_{max}$

and h_0 *divides* b_1, *if and only if* $\delta_1 h_0 \le m$.

Proof. If h_0 divides b_1, then $\delta_1 h_0 \le \delta_1 b_1 \le m$; this proves the "only if"-part.

We prove the "if"-part. Suppose that $\delta_1 h_0 \le m$. The polynomial h_0 is a divisor of f, so that

$$(h_0)_{max} \le e^{\sum_{i=1}^{t} n_i}\, f_{max}$$

according to [2]. With $\delta_1 h_0 \le m$ and $\delta_i h_i \le n_i$ for $2 \le i \le t$ we get

$$|h_0| \le M^{\frac{1}{2}}\, e^{\sum_{i=1}^{t} n_i}\, f_{max},$$

so that [3: (1.11)] combined with $h_0 \in L$ (this follows from $\delta_1 h_0 \le m$) yields

$$|b_1| \le (M\,2^{M-1})^{\frac{1}{2}}\, e^{\sum_{i=1}^{t} n_i}\, f_{max}.$$

This proves (2.11) because $(b_1)_{max} \le |b_1|$. With (2.10) and (2.3) we now have that $\gcd(f,b_1) \ne 1$. Suppose that h_0 does not divide $r = \gcd(f,b_1)$. Then $\tilde{h}$ divides $\tilde{f}/\tilde{r}$, so that, with

$$(f/r)_{max} \le e^{\sum_{i=1}^{t} n_i}\, f_{max},$$

and (2.10), (2.11), and (2.3), we get that $\gcd(f/r,b_1) \ne 1$. This is a contradiction with $r = \gcd(f,b_1)$, because f does not contain multiple factors. □

(2.12) Suppose that f does not contain multiple factors and that f is primitive. Let $s_2, s_3, \ldots, s_t$ and $\tilde{h}$ be chosen such that (2.10) with m replaced by $n-1$ and (2.1) are satisfied. The divisor h_0 of f can be

determined in the following way.

For the values $m = \delta_1\tilde{n}, \delta_1\tilde{n}+1, \ldots, n-1$ in succession we apply the basis reduction algorithm (cf. [3: Section 1]) to the lattice L as defined in (2.2). We stop as soon as a vector b_1 is found satisfying (2.11). It is not difficult to see that the first vector b_1 satisfying (2.11) that we encounter, also satisfies $b_1 = \pm h_0$ (here we apply [3: (1.37)] and (2.9)). If no vector satisfying (2.11) is found, then $\delta_1 h_0 > n-1$, so that $h_0 = f$; this follows from (2.9).

(2.13) Proposition. *Assume that the conditions in* (2.12) *are satisfied. The polynomial* h_0 *can be computed in* $O((\delta_1 h_0 N_2)^4 \log B)$ *arithmetic operations on integers having binary length* $O(N \log B)$, *where*

$$\log B = O(\log f_{max} + n + \log N_2 + \Sigma_{i=2}^{t} n_i \log s_i).$$

Proof. Combining

$$|\tilde{n}| \leq \binom{2n}{n}^{\frac{1}{2}} |f|$$

(cf. [7]) and (2.7), we find that

$$|\tilde{n}| \leq f_{max}((n+1)\binom{2n}{n})^{\frac{1}{2}} \Pi_{i=2}^{t} (n_i+1) s_i^{n_i}.$$

The proof follows now immediately from (2.2), [3: (1.26)] and [3: (1.37)]. □

(2.14) We describe an algorithm to compute the irreducible factors of f in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$. Assume that f is primitive.

First we compute the resultant $R = R(f,f') \in \mathbb{Z}[X_2, X_3, \ldots, X_t]$ of f and its derivative f' with respect to X_1, using the subresultant algorithm from [1]. We may assume that $R \neq 0$, i.e. f does not contain multiple

factors. (In the case that $R = 0$, the greatest common divisor g of f and f' is also computed by the subresultant algorithm, and the factoring algorithm can be applied to f/g.)

Next we determine $s_2, s_3, \ldots, s_t \in \mathbb{Z}$ such that $\tilde{R} \neq 0$ and such that (2.10) is satisfied with m replaced by $n-1$:

$$(2.15) \qquad s_j \geq (n N_2 2^{nN_2-1})^{n/2} (2n-1)! \, (e^{\sum_{i=1}^{t} n_i} f_{max}^{N_2} \prod_{i=2}^{j-1} s_i^{n_i})^{2n-1}$$

for $2 \leq j \leq t$. It follows from the reasoning in the proof of (2.3) that if we take $s_j \in \mathbb{Z}_{>0}$ minimal such that (2.15) is satisfied, then $\tilde{R} \neq 0$.

By means of the algorithm from [3] we compute the irreducible and primitive factors of f of degree > 0 in X_1. The condition $\tilde{R} \neq 0$ implies that (2.1) holds for every irreducible factor $\tilde{h}$ of $\tilde{f}$ thus found.

Finally, the factorization of f is determined by repeated application of the algorithm described in (2.12).

(2.16) Theorem. *Let f be a polynomial in $\mathbb{Z}[X_1, X_2, \ldots, X_t]$ with $t \geq 2$, $\delta_i f = n_i$, and $2 \leq n = n_1 \leq n_2 \leq \ldots \leq n_t$. The irreducible factorization of f can be found in $O(n^{t-2}(N^6 + N^5 \log f_{max}))$ arithmetic operations on integers having binary length $O(n^{t-2}(N^3 + N^2 \log f_{max}))$, where $N = \prod_{i=1}^{t}(n_i+1)$.*

Remark. Because $n^t = O(N)$, Theorem (2.16) implies that f can be factored in time polynomial in N and $\log f_{max}$.

Proof of (2.16). First assume that f is primitive. The resultant R can be computed in $O(n^{3t-1}N_2^2)$ arithmetic operations on integers having binary length $O(n^2 \log(f_{max}N_2))$ (cf. [1]).

The cost of applying (2.12) therefore dominates the costs of the computation of R and the factorization of $\tilde{f}$.

The same estimates are valid in the case that $R = 0$. In this case we have that

$$(f/g)_{max} \leq e^{\sum_{i=1}^{t} n_i} f_{max}$$

(cf. [2]), so that the same estimates as above are valid for the computation of the factorization of f/g.

Finally, we consider the case that the content of f is unequal to one. The computation of $cont(f)$ can be done in $O(n n_2^{3t-4} N_3^2)$ arithmetic operations on integers having binary length $O(n_2^2 \log(f_{max} N_3))$ (cf. [1]). Because $\delta_i f = \delta_i cont(f) + \delta_i (f/cont(f))$ for $2 \leq i \leq t$, the proof follows by repeated application of the above reasoning. □

(2.19) Remark. As mentioned in the introduction, a somewhat more complicated but similar approach leads to an algorithm that does not depend on the polynomial-time algorithm for factoring in $\mathbb{Z}[X]$. Instead, it can be seen as a direct generalization of the $\mathbb{Z}[X]$-algorithm. We will not give a detailed description of this alternative method here, we only indicate the main differences.

The divisor $\tilde{h} \in \mathbb{Z}[X_1]$ of $\tilde{f}$ is replaced by a divisor $(\tilde{h} \bmod p^k) \in (\mathbb{Z}/p^k\mathbb{Z})[X_1]$ of $(\tilde{f} \bmod p^k)$, for some suitably chosen prime power p^k. Condition (2.2)(ii) is therefore replaced by the condition that $(\tilde{h} \bmod p^k)$ divides $(\tilde{g} \bmod p^k)$ in $(\mathbb{Z}/p^k\mathbb{Z})[X_1]$. The lattice $L \subset \mathbb{Z}^M$ now has rank M, and a basis for L is given by

$$\{p^k X_1^i : \ 0 \leq i < \delta_1 \tilde{h}\}$$

From the choice of s_j (cf. (2.15)) we derive

$$\log s_j = O(n^2 N_2 + n \log f_{max} + \Sigma_{i=2}^{j-1} n\, n_i \log s_i)$$

for $2 \le j \le t$, so that

$$\log s_j = O((n^2 N_2 + n \log f_{max}) \Pi_{i=2}^{j-1} (1+n\, n_i)).$$

This yields

$$\Sigma_{i=2}^{t} n_i \log s_i = O(n^{t-2}(N^2 + N \log f_{max})), \tag{2.17}$$

which gives, combined with (2.7),

$$\log \bar{f}_{max} = O(n^{t-2}(N^2 + N \log f_{max})). \tag{2.18}$$

The polynomial $\bar{f}$ can be factored in $O(n^6 + n^5 \log \bar{f}_{max})$ arithmetic operations on integers having binary length $O(n^3 + n^2 \log \bar{f}_{max})$, according to [3: (3.6)]. With (2.18) this becomes

$$O(n^{t+3}(N^2 + N \log f_{max}))$$

arithmetic operations on integers having binary length

$$O(n^t(N^2 + N \log f_{max})).$$

According to (2.13) and (2.17), repeated application of the algorithm described in (2.12) takes

$$O(n^{t-2}(N^6 + N^5 \log f_{max}))$$

arithmetic operations on integers having binary length

$$O(n^{t-2}(N^3 + N^2 \log f_{max})).$$

$$\cup \{(\bar{h} \bmod p^k) x_1^{i-\delta_1 \bar{h}}: \ \delta_1 \bar{h} \le i \le m\}$$

$$\cup \{x_1^i \Pi_{j=2}^t (x_j - s_j)^{i_j}: \ 0 \le i \le m, \ 0 \le i_j \le n_j \text{ for } 2 \le j \le t, \text{ and}$$

$$(i_2, i_3, \ldots, i_t) \ne (0, 0, \ldots, 0)\}.$$

Again, it can be proven that, if $s_2, s_3, \ldots, s_t$ and p^k are sufficiently large, then the irreducible factor of f that corresponds to $(\bar{h} \bmod p^k)$ is the shortest vector in L. This factor can therefore be found by means of the basis reduction algorithm, and the resulting algorithm appears to be polynomial-time. For $f \in \mathbb{Z}[X,Y]$ the details are given in [5], and for $f \in \mathbb{Q}(\alpha)[X_1, X_2, \ldots, X_t]$ in [6].

References.

1. W.S. Brown, The subresultant PRS algorithm. ACM Transactions on mathematical software 4 (1978), 237-249.
2. A.O. Gel'fond, Transcendental and algebraic numbers, Dover Publ., New York 1960.
3. A.K. Lenstra, H.W. Lenstra, Jr., L. Lovász, Factoring polynomials with rational coefficients, Math. Ann. 261 (1982), 515-534.
4. A.K. Lenstra, Factoring multivariate polynomials over finite fields, Report IW 221/83, Mathematisch Centrum, Amsterdam 1983 (also Proceedings 15th STOC).
5. A.K. Lenstra, Factoring multivariate integral polynomials, Report IW 229/83, Mathematisch Centrum, Amsterdam 1983 (also Proceedings 10th ICALP).

6. A.K. Lenstra, Factoring multivariate polynomials over algebraic number fields, to appear.

7. M. Mignotte, An inequality about factors of polynomials, Math. Comp. 28 (1974), 1153-1157.


pp molto espr.

Factoring multivariate polynomials over algebraic number fields

by

A.K. Lenstra

ABSTRACT

We present an algorithm to factor multivariate polynomials over algebraic number fields that is polynomial-time in the degrees of the polynomial to be factored. The algorithm is an immediate generalization of the polynomial-time algorithm to factor univariate polynomials with rational coefficients.

KEY WORDS & PHRASES: *polynomial algorithm, polynomial factorization*

1. Introduction.

We show that the algorithm from [7] to factor univariate polynomials with rational coefficients can be generalized to multivariate polynomials with coefficients in an algebraic number field. As a result we get an algorithm that is polynomial-time in the degrees and the coefficient-size of the polynomial to be factored.

An outline of the algorithm is as follows. First the polynomial $f \in \mathbb{Q}(\alpha)[X_1, X_2, \ldots, X_t]$ is evaluated in a suitably chosen integer point $(X_2 = s_2, X_3 = s_3, \ldots, X_t = s_t)$. Next, for some prime number p, a p-adic irreducible factor $\tilde{h}$ of the resulting polynomial $\tilde{f} \in \mathbb{Q}(\alpha)[X_1]$ is determined up to a certain precision. We then show that the irreducible factor h_0 of f for which $\tilde{h}$ is a p-adic factor of $\tilde{h}_0$, belongs to a certain integral lattice, and that h_0 is relatively short in this lattice. This enables us to compute this factor h_0 by means of the so-called *basis reduction algorithm* (cf. [7: Section 1]).

As [7] is easily available, we do not consider it to be necessary to recall the basis reduction algorithm here; we will assume the reader to be familiar with this algorithm and its properties.

Although the algorithm presented in this paper is polynomial-time, we do not think it is a useful method for practical purposes. Like the other generalizations of the algorithm from [7], which can be found in [8;9;10; 11], the algorithm will be slow, because the basis reduction algorithm has to be applied to huge dimensional lattices with large entries. In practice, a combination of the methods from [6], [14], and [15] can be recommended (cf. [6]).

2. Preliminaries.

In this section we introduce some notation, and we derive an upper bound for the coefficients of factors of multivariate polynomials over algebraic number fields.

Let the algebraic number field $\mathbb{Q}(\alpha)$ be given as the field of rational numbers $\mathbb{Q}$ extended by a root α of a prescribed *minimal polynomial* $F \in \mathbb{Z}[T]$ with leading coefficient equal to one; i.e. $\mathbb{Q}(\alpha) \simeq \mathbb{Q}[T]/(F)$. Similarly, we define $\mathbb{Z}[\alpha] = \mathbb{Z}[T]/(F)$ as a ring of polynomials in α over $\mathbb{Z}$ of degree $< I$, where I denotes the degree δF of F.

Let $f \in \mathbb{Q}(\alpha)[X_1, X_2, \ldots, X_t]$ be the polynomial to be factored, with the number of variables $t \geq 2$. By $\delta_i f = n_i$ we denote the degree of f in X_i, for $1 \leq i \leq t$. We often use n instead of n_1. We put $N_i = \prod_{k=i}^{t} (n_k+1)$, and $N = N_1$. Let $\ell c_0(f) = f$. For $1 \leq i \leq t$ we define $\ell c_i(f) \in \mathbb{Q}(\alpha)[X_{i+1}, X_{i+2}, \ldots, X_t]$ as the leading coefficient with respect to X_i of $\ell c_{i-1}(f)$, and we put $\ell c(f) = \ell c_t(f)$. Finally, we define the *content* $\mathrm{cont}(f) \in \mathbb{Q}(\alpha)[X_2, X_3, \ldots, X_t]$ of f as the greatest common divisor of the coefficients of f with respect to X_1. Without loss of generality we may assume that $2 \leq n_i \leq n_{i+1}$ for $1 \leq i < t$, that f is *monic* (i.e. $\ell c(f) = 1$), and that $\delta_i \mathrm{cont}(f) = 0$ for $2 \leq i \leq t$.

Let $d \in \mathbb{Z}_{>0}$ be such that $f \in \frac{1}{d}\mathbb{Z}[\alpha][X_1, X_2, \ldots, X_t]$, and let $\mathrm{discr}(F)$ denote the discriminant of F. It is well-known (cf. [15]) that if we take $D = d|\mathrm{discr}(F)|$, then all monic factors of f are in $\frac{1}{D}\mathbb{Z}[\alpha][X_1, X_2, \ldots, X_t]$ (in fact it is sufficient to take $D = ds$, where s is the largest integer such that s^2 divides $\mathrm{discr}(F)$, but this integer s might be too difficult to compute).

We now introduce some notation, similar to [8: Section 1]. Suppose that we are given a prime number p such that

(2.1) p does not divide D.

For $G = \Sigma_i a_i T^i \in \mathbb{Z}[T]$ we denote by G_ℓ or $G \bmod p^\ell$ the polynomial $\Sigma_i (a_i \bmod p^\ell) T^i \in (\mathbb{Z}/p^\ell\mathbb{Z})[T]$, for any positive integer ℓ. Suppose furthermore that we are given some positive integer k, and that p is chosen in such a way that a polynomial $H \in \mathbb{Z}[T]$ exists such that

(2.2) H has leading coefficient equal to one,

(2.3) H_k divides F_k in $(\mathbb{Z}/p^k\mathbb{Z})[T]$,

(2.4) H_1 is irreducible in $(\mathbb{Z}/p\mathbb{Z})[T]$,

(2.5) $(H_1)^2$ does not divide F_1 in $(\mathbb{Z}/p\mathbb{Z})[T]$.

Clearly H_1 divides F_1 in $(\mathbb{Z}/p\mathbb{Z})[T]$, and $0 < \delta H \leq I$. In the sequel we will assume that conditions (2.1), (2.2), (2.3), (2.4), and (2.5) are satisfied.

By $\mathbb{F}_q$ we denote the finite field containing $q = p^{\delta H}$ elements. From (2.4) we have $\mathbb{F}_q \simeq (\mathbb{Z}/p\mathbb{Z})[T]/(H_1) \simeq \{\Sigma_{i=0}^{\delta H - 1} a_i \alpha_1^i : a_i \in \mathbb{Z}/p\mathbb{Z}\}$, where $\alpha_1 = T \bmod (H_1)$ is a zero of H_1. Furthermore we put $W_k(\mathbb{F}_q) = (\mathbb{Z}/p^k\mathbb{Z})[T]/(H_k) = \{\Sigma_{i=0}^{\delta H - 1} a_i \alpha_k^i : a_i \in \mathbb{Z}/p^k\mathbb{Z}\}$, where $\alpha_k = T \bmod (H_k)$ is a zero of H_k. Notice that $W_k(\mathbb{F}_q)$ is a ring containing q^k elements, and that $W_1(\mathbb{F}_q) \simeq \mathbb{F}_q$. For $a \in \mathbb{Z}[\alpha]$ we denote by $a \bmod (p^\ell, H_\ell) \in W_\ell(\mathbb{F}_q)$ the result of the canonical mapping from $\mathbb{Z}[\alpha] = \mathbb{Z}[T]/(F)$ to $W_\ell(\mathbb{F}_q) = (\mathbb{Z}/p^\ell\mathbb{Z})[T]/(H_\ell)$ applied to a, for $\ell = 1, k$. For $\tilde{g} = \Sigma_i \frac{a_i}{D} X_1^i \in \frac{1}{D}\mathbb{Z}[\alpha][X_1]$ we denote by $\tilde{g} \bmod (p^\ell, H_\ell)$ the polynomial $\Sigma_i (((D^{-1} \bmod p^\ell)\, a_i) \bmod (p^\ell, H_\ell))\, X_1^i \in W_\ell(\mathbb{F}_q)[X_1]$ (notice that $D^{-1} \bmod p^\ell$ exists due to (2.1)).

We derive an upper bound for the height of a monic factor g of f. As usual, for $g = \sum_{i_1} \sum_{i_2} \dots \sum_{i_t} \sum_{j} a_{i_1 i_2 \dots i_t j} \alpha^j x_1^{i_1} x_2^{i_2} \dots x_t^{i_t} \in \mathbb{Q}(\alpha)[x_1, x_2, \dots, x_t]$, the *height* g_{max} is defined as $\max|a_{i_1 i_2 \dots i_t j}|$, and the *length* $|g|$ as $(\Sigma a^2_{i_1 i_2 \dots i_t j})^{\frac{1}{2}}$. Similarly, for a polynomial h with complex coefficients, we define its height h_{max} as the maximum of the absolute values of its complex coefficients.

For any choice of $\alpha \in \{\alpha_1, \alpha_2, \dots, \alpha_I\}$, where $\alpha_1, \alpha_2, \dots, \alpha_I$ are the conjugates of α, we can regard g as a polynomial g_α with complex coefficients. We define $||g||$ as $\max_{1 \le i \le I} (g_{\alpha_i})_{max}$. From [3] we have

$$||g|| \le e^{\Sigma_{i=1}^{t} n_i} ||f||.$$

In [8: Section 4] we have shown that this leads to

$$g_{max} \le e^{\Sigma_{i=1}^{t} n_i} ||f|| \, I \, (I-1)^{(I-1)/2} |F|^{I-1} |\mathrm{discr}(F)|^{-\frac{1}{2}}. \tag{2.6}$$

From [13] we know that the length $|F|$ of F is an upper bound for the absolute value of the conjugates of α, so that

$$||f|| \le f_{max} \Sigma_{i=0}^{I-1} |F|^i,$$

which yields, combined with (2.6),

$$g_{max} \le e^{\Sigma_{i=1}^{t} n_i} f_{max} \, I \, (I-1)^{(I-1)/2} |F|^{I-1} |\mathrm{discr}(F)|^{-\frac{1}{2}} \Sigma_{i=0}^{I-1} |F|^i. \tag{2.7}$$

The upper bound for the height of monic factors of f, as given by the right hand side of (2.7), will be denoted by B_f. Because $|\mathrm{discr}(F)| \ge 1$, we find

$$\log B_f = O(\Sigma_{i=1}^{t} n_i + \log f_{max} + I \log(I|F|)). \tag{2.8}$$

3. Factoring multivariate polynomials over algebraic number fields.

We describe an algorithm to compute the irreducible factorization of f in $\mathbb{Q}(\alpha)[X_1, X_2, \ldots, X_t]$.

Let $s_2, s_3, \ldots, s_t \in \mathbb{Z}_{>0}$ be a $(t-1)$-tuple of integers. For $g \in \mathbb{Q}(\alpha)[X_1, X_2, \ldots, X_t]$ we denote by $\tilde{g}_j$ the polynomial g modulo $((X_2-s_2), (X_3-s_3), \ldots, (X_j-s_j)) \in \mathbb{Q}(\alpha)[X_1, X_{j+1}, X_{j+2}, \ldots, X_t]$; i.e. $\tilde{g}_j$ is g with s_i substituted for X_i, for $2 \le i \le j$. Notice that $\tilde{g}_1 = g$ and that $\tilde{g}_j = \tilde{g}_{j-1}$ modulo $(X_j - s_j)$. We put $\tilde{g} = \tilde{g}_t$.

Suppose that a polynomial $\tilde{h} \in \mathbb{Z}[\alpha][X_1]$ is given such that

(3.1) $\tilde{h}$ is monic,

(3.2) $\tilde{h} \bmod (p^k, H_k)$ divides $\tilde{f} \bmod (p^k, H_k)$ in $W_k(\mathbb{F}_q)[X_1]$,

(3.3) $\tilde{h} \bmod (p, H_1)$ is irreducible in $\mathbb{F}_q[X_1]$,

(3.4) $(\tilde{h} \bmod (p, H_1))^2$ does not divide $\tilde{f} \bmod (p, H_1)$ in $\mathbb{F}_q[X_1]$.

We put $\ell = \delta_1 \tilde{h}$, so $0 < \ell \le n$. By $h_0 \in \frac{1}{D}\mathbb{Z}[\alpha][X_1, X_2, \ldots, X_t]$ we denote the unique, monic, irreducible factor of f such that $\tilde{h} \bmod (p^k, H_k)$ divides $\tilde{h}_0 \bmod (p^k, H_k)$ in $W_k(\mathbb{F}_q)[X_1]$ (cf. (3.2), (3.3), (3.4)).

(3.5) Let $m = m_1, m_2, m_3, \ldots, m_t$ be a t-tuple of integers satisfying $\ell \le m < n$ and $0 \le m_i \le \delta_i \ell c_{i-1}(f)$ for $2 \le i \le t$, and let $M = 1 + I\Sigma_{i=1}^{t} m_i N_{i+1}$ (where of course $N_{t+1} = 1$). We define $L \subset (\frac{\mathbb{Z}}{D})^M$ as the lattice of rank M, consisting of the polynomials $g \in \frac{1}{D}\mathbb{Z}[\alpha][X_1, X_2, \ldots, X_t]$ for which

(i) $\delta_1 g \le m$ and $\delta_i g \le n_i$ for $2 \le i \le t$;

(ii) If $\delta_j \ell c_{j-1}(g) = m_j$ for $1 \le j \le i$, then $\delta_{i+1} \ell c_i(g) \le m_{i+1}$ for $1 \le i < t$;

(iii) If $\delta_i \ell c_{i-1}(g) = m_i$ for $1 \le i \le t$, then $\ell c(g) \in \mathbb{Z}$;

(iv) $\tilde{n} \bmod (p^k, H_k)$ divides $\tilde{g} \bmod (p^k, H_k)$ in $W_k(\mathbb{F}_q)[X_1]$.

Here M-dimensional vectors and polynomials satisfying conditions (i), (ii), and (iii), are identified in the usual way (cf. [8: (2.6); 11: (2.2)]). For notational convenience we only give a basis for L in the case that $m_i = n_i$ for $2 \le i \le t$; the general case can easily be derived from this:

$$\{\tfrac{1}{D} p^k \alpha^j X_1^i : \ 0 \le j < \delta H, \ \ 0 \le i < \ell\}$$

$$\cup \{\tfrac{1}{D} \alpha^{j-\delta H} H(\alpha) X_1^i : \ \delta H \le j < I, \ \ 0 \le i < \ell\}$$

$$\cup \{\tfrac{1}{D} \alpha^j \tilde{n} X_1^{i-\ell} : \ 0 \le j < I, \ \ \ell \le i \le m\}$$

$$\cup \{\tfrac{1}{D} \alpha^j X_1^{i_1} \Pi_{r=2}^t (X_r - s_r)^{i_r} : \ 0 \le j < I, \ \ 0 \le i_1 \le m, \ \ 0 \le i_r \le n_r$$
$$\text{for } 2 \le r \le t, \ \ (i_2, i_3, \ldots, i_t) \ne (0, 0, \ldots, 0),$$
$$\text{and } (i_1, i_2, i_3, \ldots, i_t) \ne (m, n_2, n_3, \ldots, n_t)\}$$

$$\cup \{X_1^m \Pi_{r=2}^t (X_r - s_r)^{n_r}\}$$

(cf. [8: (2.6); 11: (2.19)], (2.2), and (3.1)).

(3.6) Proposition. *Let* b *be a non-zero element of* L *and let*

$$\tilde{B}_j = f_{max}^m b_{max}^n (n+m)! \left(D N_2 (1+F_{max})^{I-1} \Pi_{i=2}^j s_i^{n_i} \right)^{n+m}, \tag{3.7}$$

for $1 \le j \le t$, *where* f_{max}^m *denotes* $(f_{max})^m$.

Suppose that

$$s_j \ge ((n+m) n_j + 1)^{\frac{1}{2}} \tilde{B}_{j-1} \tag{3.8}$$

for $2 \le j \le t$, *and*

$$(3.9) \qquad p^{k\delta H} \ge |F|^{I-1} (I^{\frac{1}{2}} \tilde{B}_t)^I .$$

Then $\gcd(f,b) \neq 1$ *in* $\mathbb{Q}(\alpha)[X_1, X_2, \ldots, X_t]$.

Proof. Denote by $R = R(Df,Db) \in \mathbb{Z}[\alpha][X_2, X_3, \ldots, X_t]$ the resultant of Df and Db (with respect to the variable X_1). An outline of the proof is as follows. First we prove that an upper bound for $(\tilde{R}_j)_{max}$ is given by $\tilde{B}_j$. Combining this with (3.8), we then see that $X_j = s_j$ cannot be a zero of $\tilde{R}_{j-1}$ if $\tilde{R}_{j-1} \neq 0$, for $2 \le j \le t$. This implies that the assumption that $R \neq 0$ (i.e. $\gcd(f,b) = 1$) leads to $\tilde{R} \neq 0$. We then apply a result from [6], and we find with (3.9) that $\tilde{R} \bmod (p^k, H_k) \neq 0$. But this is a contradiction, because $\tilde{h} \bmod (p^k, H_k)$ divides both $\tilde{f} \bmod (p^k, H_k)$ and $\tilde{b} \bmod (p^k, H_k)$ in $W_k(\mathbb{F}_q)[X_1]$. We conclude that $R = 0$, so that $\gcd(f,b) \neq 1$ in $\mathbb{Q}(\alpha)[X_1, X_2, \ldots, X_t]$.

If a and b are two polynomials in any number of variables over $\mathbb{Q}(\alpha)$, having ℓ_a and ℓ_b terms respectively, then

$$(3.10) \qquad (ab)_{max} \le a_{max}\, b_{max} \min(\ell_a, \ell_b)\, (1 + F_{max})^{I-1} .$$

From (3.10) we easily derive an upper bound for $(\tilde{R}_j)_{max}$, because $\tilde{R}_j \in \mathbb{Z}[\alpha][X_{j+1}, X_{j+2}, \ldots, X_t]$ is the resultant of $D\tilde{f}_j$ and $D\tilde{b}_j$:

$$(3.11) \qquad (\tilde{R}_j)_{max} \le (D\tilde{f}_j)^m_{max} (D\tilde{b}_j)^n_{max} (n+m)! N_{j+1}^{n+m-1} (1 + F_{max})^{(I-1)(n+m-1)} .$$

It follows from $\tilde{f}_j = \tilde{f}_{j-1} \bmod (X_j - s_j)$, that $(\tilde{f}_j)_{max} \le (\tilde{f}_{j-1})_{max} (n_j+1) s_j^{n_j}$, so that

$$(3.12) \qquad (\tilde{f}_j)_{max} \le f_{max} \prod_{i=2}^{j} (n_i+1) s_i^{n_i} .$$

Combining (3.11), (3.12), and a similar bound for $(\tilde{B}_j)_{max}$, we obtain

$$(3.13) \qquad (\tilde{R}_j)_{max} < f_{max}^m b_{max}^n (n+m)!(D N_2 \Pi_{i=2}^{j} s_i^{n_i})^{n+m} (1 + F_{max})^{(I-1)(n+m-1)},$$

for $1 \le j < t$. (Remark that (3.13) with "<" replaced by "≤" holds for $j = t$.)

Now assume, for some j with $2 \le j \le t$, that $\tilde{R}_{j-1}$ is unequal to zero. We prove that $\tilde{R}_j \neq 0$. Because $\tilde{R}_j = \tilde{R}_{j-1}$ modulo $(X_j - s_j)$, the condition $\tilde{R}_j = 0$ would imply that all polynomials in $\mathbb{Z}[X_j]$ that result from $\tilde{R}_{j-1}$ by grouping together all terms with identical exponents in α and X_{j+1} up to X_t, have $(X_j - s_j)$ as a factor. These polynomials have degree (in X_j) at most $(n+m)n_j$, so that we get, with the result from [12], that

$$|s_j| \le ((n+m)n_j+1)^{\frac{1}{2}} (\tilde{R}_{j-1})_{max}.$$

Combined with (3.13) and (3.7) this is a contradiction with (3.8). We conclude that $\tilde{R}_j \neq 0$ if $\tilde{R}_{j-1} \neq 0$ for any j with $2 \le j \le t$, so that the assumption $\gcd(f,b) = 1$ (i.e. $R \neq 0$) leads to $\tilde{R} \neq 0$.

Assume that $H_k(T)$ divides $\tilde{R}(T) \in \mathbb{Z}[T]$ in $(\mathbb{Z}/p^k\mathbb{Z})[T]$, i.e. $\tilde{R} \bmod (p^k, H_k) = 0$. The polynomial $H_k(T)$ is also a divisor of $F(T)$ in $(\mathbb{Z}/p^k\mathbb{Z})[T]$, so that $\gcd(F(T), \tilde{R}(T)) = 1$ and [6: Theorem 2] lead to

$$p^{k\delta H} \le |F|^{I-1} (I^{\frac{1}{2}} \tilde{R}_{max})^I.$$

With the remark after (3.13) and (3.7) this is a contradiction with (3.9), so that $\tilde{R} \bmod (p^k, H_k) \neq 0$. This concludes the proof of (3.6). □

(3.14) Proposition. *Let* $b_1, b_2, \ldots, b_M$ *be a reduced basis for* L *(cf.* [7: Section 1]*), where* L *and* M *are as in* (3.5)*, and let*

$$(3.15)\qquad B_j = (n+m)!\,(M2^{M-1})^{n/2}\left(B_f\, D\, N_2 (1+F_{max})^{I-1} \prod_{i=2}^{j} s_i^{n_i}\right)^{n+m},$$

for $2 \le j \le t$, *where* B_f *is as in Section 2. Suppose that*

$$(3.16)\qquad s_j \ge ((n+m)n_j+1)^{\frac{1}{2}} B_{j-1}$$

for $2 \le j \le t$, *that*

$$(3.17)\qquad p^{k\delta H} \ge |F|^{I-1} (I^{\frac{1}{2}} B_t)^I,$$

and that f *does not contain multiple factors. Then*

$$(3.18)\qquad (b_1)_{max} \le (M2^{M-1})^{\frac{1}{2}} B_f$$

and h_0 *divides* b_1, *if and only if* $h_0 \in L$.

Proof. If h_0 divides b_1, then $h_0 \in L$, because $b_1 \in L$; this proves the "if"-part.

To prove the "only if"-part, suppose that $h_0 \in L$. Because h_0 is a monic factor of f, we have from (2.7) that $(h_0)_{max} \le B_f$. With [7: (1.11)] and $h_0 \in L$ this gives $|b_1| \le (M2^{M-1})^{\frac{1}{2}} B_f$ so that (3.18) holds, because $(b_1)_{max} \le |b_1|$. Because of (3.18), (3.16), (3.17), (3.15), and the definition of B_f, we can apply (3.6), which yields $\gcd(f,b_1) \ne 1$.

Now suppose that h_0 does not divide b_1. This implies that h_0 also does not divide $r = \gcd(f,b_1)$, where r can be assumed to be monic. But then $\bar{h} \bmod (p^k, H_k)$ divides $(\bar{f}/\bar{r}) \bmod (p^k, H_k)$, so that Proposition (3.6) can be applied with f replaced by f/r. Conditions (3.8) and (3.9) are satisfied because $(f/r)_{max} \le B_f$ (cf. (2.7)) and because of (3.16), (3.17), and (3.15). It follows that $\gcd(f/r,b_1) \ne 1$, which contradicts $r = \gcd(f,b_1)$ because f does not contain multiple factors. □

(3.19) We describe how to compute the irreducible factor h_0 of f. Suppose that f does not contain multiple factors, and that the polynomial $\bar{h}$, the $(t-1)$-tuple $s_2, s_3, \ldots, s_t$, and the prime power p^k are chosen such that (3.1), (3.2), (3.3), (3.4), (3.16), and (3.17) are satisfied with, for (3.16) and (3.17), m replaced by $n-1$. Remember that we also have to take care that conditions (2.1), (2.2), (2.3), (2.4), and (2.5) on p and H are satisfied.

We apply the basis reduction algorithm (cf. [7: Section 1]) to a sequence of M_j-dimensional lattices as in (3.5), where the $M_j = 1 + I\Sigma_{i=1}^{t} m_i N_{i+1}$ run through the range of admissible values for $m_1, m_2, \ldots, m_t$ (cf. (3.5)), in such a way that $M_j < M_{j+1}$. (So, for $m = \ell, \ell+1, \ldots, n-1$, and $m_i = 0, 1, \ldots, \delta_i {}^{\ell}c_{i-1}(f)$ for $i = t, t-1, \ldots, 2$ in succession.) According to (3.14), the first vector b_1 that we find that satisfies (3.18) equals $\pm h_0$ (remember that b_1 belongs to a basis for the lattice), so that we can stop if such a vector is found. If for none of the lattices a vector satisfying (3.18) is found, then h_0 is not contained in any of these lattices according to (3.14), so that $h_0 = f$.

(3.20) Proposition. *Assume that the conditions in* (3.19) *are satisfied. The polynomial* h_0 *can be computed in* $O((\delta_1 h_0 I N_2)^4 k \log p)$ *arithmetic operations on integers having binary length* $O(I N k \log p)$.

Proof. Observing that $\log(I N p^{2k}) = O(k \log p)$ (cf. (3.17), (3.15), and (2.8)), the proof immediately follows from (3.19), (3.5), and [7: (1.26), (1.37)]. □

(3.21) We now show how $s_2, s_3, \ldots, s_t$ and p can be chosen in such a way that the conditions in (3.19) can be satisfied. The algorithm to factor f then easily follows by repeated application of (3.19).

We assume that f does not contain multiple factors, so that the resultant $R = R(df,df')$ of df and its derivative df' with respect to X_1 is unequal to zero. First we choose $s_2, s_3, \ldots, s_t \in \mathbb{Z}_{>0}$ minimal such that (3.16) is satisfied with m replaced by $n-1$. It follows from (3.16), (3.15), (2.8), and $\log D = O(\log d + I \log(I|F|))$ (because $D = d|\mathrm{discr}(F)|$), that

$$\begin{aligned} \log s_j &= O(\log((n+m)n_j) + \log B_{j-1}) \\ &= O(I\,n\,N + n(\log B_f + \log D + I \log(1+F_{max}) + \Sigma_{i=1}^{j-1} n_i \log s_i)) \\ &= O(n(I\,N + \log(df_{max}) + I \log(I|F|) + \Sigma_{i=1}^{j-1} n_i \log s_i)) \end{aligned}$$

for $2 \le j \le t$, so that

$$\log s_j = O(n(I\,N + \log(df_{max}) + I \log(I|F|)) \prod_{i=2}^{j-1}(1+n\,n_i))$$

and

$$\Sigma_{i=2}^{t} n_i \log s_i = O(n^{t-2} N(I\,N + \log(df_{max}) + I \log(I|F|))). \tag{3.22}$$

From the proof of (3.6) it follows that, for this choice of $s_2, s_3, \ldots, s_t$ the resultant $R \in \mathbb{Z}[\alpha]$ of df and df' is unequal to zero.

Next we choose p minimal such that p does not divide D or $\mathrm{discr}(F)$, and such that $\tilde{R} \not\equiv 0$ modulo p. Clearly

$$\prod_{q \text{ prime}, q<p} q \le d\,\mathrm{discr}(F)\,\tilde{R}_{max}$$

which yields, together with

$$\prod_{q \text{ prime}, q<p} q > e^{Ap}$$

for all $p > 2$ and some constant $A > 0$ (cf. [4: Section 22.2]), that

$$(3.23) \qquad p = O(\log d + I \log(I|F|) + \log \tilde{R}_{max}).$$

Similar to (3.13) we obtain

$$\tilde{R}_{max} \leq f_{max}^{2n-1} n^n (2n-1)! (d N_2 \prod_{i=2}^{t} s_i^{n_i})^{2n-1} (1 + F_{max})^{(I-1)(2n-2)},$$

so that we get, using (3.22)

$$\log \tilde{R}_{max} = O(n^{t-1} N(I N + \log(d f_{max}) + I \log(I|F|))).$$

Combining this with (3.23) we conclude that

$$(3.24) \qquad p = O(n^{t-1} N(I N + \log(d f_{max}) + I \log(I|F|))).$$

Notice that (2.1) is now satisfied. In order to compute a polynomial $H \in \mathbb{Z}[T]$ satisfying (2.2), (2.4), (2.5), and (2.3) with k replaced by 1, we factor $F \bmod p$ by means of Berlekamp's algorithm [5: Section 4.6.2] and we choose H as an irreducible factor of $F \bmod p$ for which $\tilde{R} \bmod (p, H_1) \neq 0$; such a polynomial H exists because $\tilde{R} \bmod p \neq 0$. Conditions (2.4) and (2.3) with k replaced by 1 are clear from the construction of H, and because we may assume that H has leading coefficient equal to one, (2.2) also holds. The condition that $\mathrm{discr}(F) \bmod p \neq 0$, finally, guarantees that $F \bmod p$ does not contain multiple factors, so that (2.5) is satisfied.

We choose k minimal such that (3.17) holds, so that

$$k \log p = O(I(I n N + n \log(d f_{max}) + I n \log(I|F|) + n \sum_{i=2}^{t} n_i \log s_i) + \log p)$$

(cf. (3.15) and (2.8)), which gives, with (3.22) and (3.24)

$$(3.25) \qquad k \log p = O(I n^{t-1} N(I N + \log(d f_{max}) + I \log(I|F|))).$$

Now we apply Hensel's lemma [5: Exercise 4.6.22] to modify H in such a way that (2.3) holds for this value of k (this is possible because (2.3) already holds for $k = 1$), and finally we apply Berlekamp's algorithm as described in [1: Section 5] and Hensel's lemma as in [14] to compute the irreducible factorization of $\tilde{f} \bmod (p^k, H_k)$ in $W_k(\mathbb{F}_q)[X_1]$. Condition (3.4) is satisfied for each irreducible factor $\tilde{h} \bmod (p^k, H_k)$ of $\tilde{f} \bmod (p^k, H_k)$ because $\tilde{R} \bmod (p, H_1) \neq 0$, and (3.1), (3.2), and (3.3) are clear from the construction of $\tilde{h}$.

We have shown how to choose $s_2, s_3, \ldots, s_t$ and p, and how to satisfy the conditions in (3.19). We are now ready for our theorem.

(3.26) Theorem. *Let* f *be a monic polynomial in* $\frac{1}{d}\mathbb{Z}[\alpha][X_1, X_2, \ldots, X_t]$ *with* $t \geq 2$, *of degree* n_i *in* X_i, *and* $2 \leq n = n_1 \leq n_2 \leq \ldots \leq n_t$. *The irreducible factorization of* f *can be found in* $O(n^{t-1}(IN)^5(IN + \log(df_{max}) + I\log(I|F|)))$ *arithmetic operations on integers having binary length* $O(n^{t-1}(IN)^2(IN + \log(df_{max}) + I\log(I|F|)))$, *where* $N = \prod_{i=1}^{t}(n_i+1)$.

Proof. If f does not contain multiple factors, then f can be factored by repeated application of (3.19). In that case (3.26) follows from (3.21). (3.20), (3.25), and the well-known estimates for the applications of Berlekamp's algorithm and Hensel's lemma (cf.[5;1] and [16]).

If f contains multiple factors, then we first have to compute the monic gcd g of f and its derivative with respect to X_1, and the factoring algorithm is then applied to f/g. The cost of factoring f/g satisfies the same estimates as above, because $(f/g)_{max} \leq B_f$ (cf. (2.7)), and this dominates the costs of the computation of g, which can be done by means of the subresultant algorithm (cf. [2]). □

References.

1. E.R. Berlekamp, Factoring polynomials over large finite fields, Math. Comp. 24 (1970), 713-735.

2. W.S. Brown, The subresultant PRS algorithm, ACM Transactions on mathematical software 4 (1978), 237-249.

3. A.O. Gel'fond, Transcendental and algebraic numbers, Dover Publ., New York 1960.

4. G.H. Hardy, E.M. Wright, An introduction to the theory of numbers, Oxford University Press 1979.

5. D.E. Knuth, The art of computer programming, vol. 2, Seminumerical algorithms, Addison-Wesley, Reading, second edition 1981.

6. A.K. Lenstra, Lattices and factorization of polynomials over algebraic number fields, Proceedings Eurocam 82, LNCS 144, 32-39.

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

8. A.K. Lenstra, Factoring polynomials over algebraic number fields, Report IW 213/82, Mathematisch Centrum, Amsterdam 1982 (also Proceedings Eurocal 83).

9. A.K. Lenstra, Factoring multivariate polynomials over finite fields, Report IW 221/83, Mathematisch Centrum, Amsterdam 1983 (also Proceedings 15th STOC, 189-192).

10. A.K. Lenstra, Factoring multivariate integral polynomials, Report IW 229/83, Mathematisch Centrum, Amsterdam 1983 (also Proceedings 10th ICALP, LNCS 154, 458-465).

11. A.K. Lenstra, Factoring multivariate integral polynomials, II, Report 230/83, Mathematisch Centrum, Amsterdam 1983.

12. M. Mignotte, An inequality about factors of polynomials, Math. Comp. 28 (1974), 1153-1157.

13. J. Stoer, Einführung in die numerische Mathematik I, Springer, Berlin 1972.

14. P.S. Wang, Factoring multivariate polynomials over algebraic number fields, Math. Comp. 30 (1976), 324-336.

15. P.J. Weinberger, L.P. Rothschild, Factoring polynomials over algebraic number fields, ACM Transactions on mathematical software 2 (1976), 335-350.

16. D.Y.Y. Yun, The Hensel lemma in algebraic manipulation, MIT, Cambridge 1974; reprint: Garland Publ. Co., New York 1980.

Samenvatting

Factorisatie van polynomen in polynomiaal begrensde tijd

Een bekende methode om polynomen met geheeltallige coëfficiënten te factoriseren is de Berlekamp-Hensel algoritme. Deze methode, die in het begin van de zeventiger jaren werd ontwikkeld, werkt in de praktijk doorgaans zeer bevredigend. Vanuit theoretisch oogpunt is de Berlekamp-Hensel algoritme evenwel minder geslaagd. Er bestaan namelijk betrekkelijk eenvoudige polynomen waarvoor het onevenredig veel tijd zou kosten de factoren met behulp van de Berlekamp-Hensel methode te bepalen. Gebruik makend van de in de complexiteitstheorie gebezigde terminologie, zeggen we dat de Berlekamp-Hensel algoritme niet in polynomiaal begrensde tijd werkt.

In dit proefschrift wordt een methode voor de factorisatie van polynomen beschreven die wel in polynomiaal begrensde tijd werkt. Dat wil zeggen dat de voor deze zogenaamde L^3-algoritme benodigde rekentijd begrensd wordt door een vaste polynomiale functie van de grootte van het te factoriseren polynoom.

De L^3-algoritme berust voornamelijk op de volgende twee observaties:

- voor ieder polynoom kan een geheeltallig rooster worden geconstrueerd, zodanig dat de korte vectoren in dit rooster aanleiding geven tot de irreducibele factoren van het polynoom;
- benaderingen van de kortste vectoren in een geheeltallig rooster kunnen worden gevonden in polynomiaal begrensde tijd.

De tweede observatie, die afkomstig is van L. Lovász, blijkt ook buiten het terrein van de factorisatie van polynomen van groot belang te zijn.

De L^3-algoritme wordt in dit proefschrift vooraf gegaan door een artikel waarin factoren van polynomen en korte vectoren in roosters voor het eerst met elkaar in verband worden gebracht. De latere artikelen bevatten generalisaties van de L^3-algoritme.

Stellingen

1. Het kleinste positieve gehele getal n met zes verschillende delers die congruent zijn modulo een gehele $s > n^{1/3}$ met $\gcd(n, s) = 1$, is 245784.
   Lit.: H.W. Lenstra, Jr., *Divisors in residue classes*, Math. Comp., 42 (1984), 331-340.

2. com sem(v, p):
   sub s: sem
   v; (v; s.v)*, (p; s.p)*
   moc.
   Deze recursief gedefinieerde semafoor levert symboolrijen af waarin het aantal v's ten minste het aantal p's is.
   Lit.: J.L.A. van de Snepscheut, *Trace theory and VLSI design*, proefschrift, Eindhoven 1983.

3. Een zogenaamde *dense encoding* is in het algemeen onbruikbaar als complexiteitsmaat voor polynomen in meer dan één veranderlijke.
   Lit.: J. Von zur Gathen, *Factoring sparse multivariate polynomials*, Proceedings 24-th annual symposium on foundations of computer science (1983), 172-179.

4. Laten b en m positieve gehele getallen zijn en laat $\tilde{\alpha} \in \mathbb{Q}(i)$ een complex rationaal getal zijn dat kan worden gerepresenteerd met s binaire bits, voor een zeker positief geheel getal $s = \Theta(m^2 + m \log b)$. De vraag of er een algebraïsch getal $\alpha \in \mathbb{C}$ bestaat met $|\alpha - \tilde{\alpha}| < 2^{-s}$ en zodanig dat het minimum polynoom $h \in \mathbb{Z}[X]$ van α L_2-norm ten hoogste b en graad ten hoogste m heeft, kan worden beantwoord in tijd polynomiaal in $\log b$ en m.
   Lit.: R. Kannan, A.K. Lenstra, L. Lovász, *Predicting bits of algebraic numbers and factorization of polynomials*, Proceedings 16-th annual ACM symposium on theory of computing (1984).

5. Zowel het vrijelijk gebruiken als het gedwongen niet gebruiken van sprongopdrachten kan leiden tot onoverzichtelijke programmatuur.
   Lit.: D.E. Knuth, *Structured computing with go to statements*, ACM computing surveys 6 (1974), 261-301.

6. Bij een informatica conferentie zijn artikelen waarvan de naam van de eerste auteur met de k-de letter van het alfabet begint, op $(\frac{2k}{26+k})$-de deel van de proceedings te verwachten, aannemende dat de bijdragen alfabetisch geordend zijn.

7. Lang zal Fortran leven.

8. Prima la musica, dopo le parole.
   Lit.: C. Krauss, R. Strauss, *Capriccio, Ein Konversationsstück für Musik in einem Aufzug*.

9. 19201518202706126283527151627010214271805110514091407142113130518275482423212
   72201142701281128271205141920180127200527011319200518040113270514270409140727
   13050527140101182704052706124428452706050519201618090513161809101 9 is priem.