Factorization of Polynomials

by E. Kaltofen

Abstract

Algorithms for factoring polynomials in one or more variables over various coefficient domains are discussed. Special emphasis is given to finite fields, the integers, or algebraic extensions of the rationals, and to multivariate polynomials with integral coefficients. In particular, various squarefree decomposition algorithms and Hensel lifting techniques are analyzed. An attempt is made to establish a complete historic trace for today's methods. The exponential worst case complexity nature of these algorithms receives attention.

Appears in *Computer Algebra*, second edition, B. Buchberger, R. Loos, G. Collins, editors, Springer Verlag, Vienna, Austria, pp. 95-11 (1982).

1. *Introduction*

The problem of factoring polynomials has a long and distinguished history. D. Knuth traces the first attempts back to Isaac Newton's Arithmetica Universalis (1707) and to the astronomer Friedrich T. v. Schubert who in 1793 presented a finite step algorithm to compute the factors of a univariate polynomial with integer coefficients. (Cf. [22, Sec. 4.6.2]) A notable criterion for determining irreducibility was given by F. G. Eisenstein in 1850 [12, p. 166]. L. Kronecker rediscovered Schubert's method in 1882 and also gave algorithms for factoring polynomials with two or more variables or with coefficients in algebraic extensions [23, Sec. 4, pp. 10-13]. Exactly one hundred years have passed since then, and though early computer programs relied on Kronecker's work [17], modern polynomial factorization algorithms and their analysis depend on major advances in mathematical research during this period of time. However, most papers which have become especially important to recent investigations do not deal with the problem per se, and we shall refer to them in the specific context.

When the long-known finite step algorithms were first put on computers they turned out to be highly inefficient. The fact that almost any uni- or multivariate polynomial of degree up to 100 and with coefficients of a moderate size (up to 100 bits) can be factored by modern algorithms in a few minutes of computer time indicates how successfully this problem has been attacked during the past fifteen years. It is the purpose of this paper to survey the methods which led to these developments. At the risk of repeating ourselves later or omitting significant contributions we shall give some main points of reference now.

In 1967 E. Berlekamp devised an ingenious algorithm which factors univariate polynomials over Z_p, p a prime number, whose running time is of order $0(n^3+prn^2)$ where n is the degree of the polynomial and r the number of actual factors (cf. [22, Sec. 4.6.2]). The incredible speed of this algorithm suggested factoring integer polynomials by first factoring them modulo certain small primes and then reconstructing the integer factors by some mean such as Chinese remaindering [21, Sec. 4.6.2]. H. Zassenhaus discussed in his landmark 1969 paper [60] how to apply the "Hensel lemma" to lift in k iterations a factorization modulo p to a factorization modulo p^{2^k}, provided that the integral polynomial is squarefree and remains squarefree modulo p. Readers familiar with basic field theory will know that if a polynomial over a field of characteristic 0 has repeated roots, then the greatest common divisor (GCD) of the polynomial and its derivative is nontrivial. Hence casting out multiple factors is essentially a polynomial GCD process, but we will come back to this problem in a later section. Squarefreeness is preserved modulo all but a reasonable small number of primes. Given a bound for the size of the coefficients of any possible polynomial factor, one then lifts the modular factorization to a factorization modulo p^{2^k} such that $p^{2^k}/2$ supersedes this coefficient bound. At this point factors with balanced residues modulo p^{2^k} either are already the integral factors or one needs to multiply some factors together to obtain a true factor over the integers. The slight complication arising from a leading coefficient not equal to unity will be resolved later.

D. Musser [32,33] and, using his ideas, P. Wang in collaboration with L. Rothschild [53], generalized the Hensel lemma to obtain factorization algorithms for multivariate integral polynomials. Subsequently P. Wang has incorporated various improvements to these multivariate factorization algorithms [49,50,52]. In 1973 J. Moses and D. Yun found the Hensel construction suitable for multivariate GCD computations (now called the EZGCD algorithm) [31], and D. Yun has used this algorithm for the squarefree decomposition process of multivariate polynomials [58]. The classical algorithm for factoring polynomials over algebraic number fields was considered and modified by B. Trager [42] but again the Hensel approach proved fruitful [48, 56]. In 1976 M. Rabin, following an idea of E. Berlekamp [5], introduced random choices in his algorithm for factoring univariate polynomials over large finite fields whose expected running time is at most of the order $0(n^3 (\log n)^3 \log(p))$ where n is the degree and p the size of the field [41], [4]. In 1979 G. Collins published a thorough analysis of the average time behavior for the univariate Berlekamp-Hensel algorithm [10], while in the same year improved algorithms for squarefree factorization [54] and Chinese remaindering on sparse multivariate polynomials appeared [63,64,65].

To completely factor a univariate polynomial over the integer means, of course, to also factor the common divisor of all its coefficients. This paper does not include the topic of factorization of integers and we will not consider this problem as a part of polynomial factorization. However, some comparisons are in order. Factoring large random integers seems much harder than factoring integral polynomials. This was partially confirmed by a polynomial-time reduction from polynomial to integer factorization, which is, however, subject to an old number theoretic conjecture [3]. The problem of finding polynomially long irreducibility proofs ("succinct certificates") was first solved for prime numbers in 1975 [40] and has recently also been achieved for densely encoded integral polynomials [7]. A polynomial-time irreducibility test for prime numbers depending on the validity of the generalized Riemann hypothesis (GRH) was discovered in 1976 (cf. [22, Sec. 4.5.4]). Peter Weinberger obtained the corresponding result for densely encoded integer polynomials [55] (also see [22, p. 632, Exercise 38]). In 1971 E. Berlekamp pointed out that the modular projection and lifting algorithm may take an exponential number of trial factor combinations [5]. Except for P. Weinberger's algorithm, whose complexity analysis is subject to the GRH, the author knows of no procedure which significantly reduces this exponential behavior in contrast to the stunning advances for the integer case by L. Adleman, C. Pomerance, and R. Rumley [2]. Also, no fast probabilistic irreducibility tests for integer polynomials seem to be known, again leaving a gap for work parallel to that of M. Rabin, R. Solovay and V. Strassen (cf. [22, Sec. 4.5.4]). Little work has been done on the theoretical analysis of the multivariate versions of the Berlekamp-Hensel algorithm. Similar to the univariate case, the steps involved may require an exponential number of trial factor combinations, though this problem may be probabilistically controllable by virtue of the Hilbert Irreducibility Theorem. G. Viry has also shown how to replace the trial divisions of multivariate polynomials by a simple degree test which makes his algorithm the asymptotically fastest, though still exponential in the degrees in the worst case, of all known deterministic algorithms [47]. Recently the author has proven that it is only polynomially harder to factor densely encoded multivariate integer polynomials with a fixed number of variables than integer polynomials with just two variables [18].

In this paper we take the concrete approach of discussing the algorithms for coefficient domains such as Galois fields, the integers, or finite algebraic extensions of the rationals. An excellent reference for a general algebraic setting is the work of D. Musser [32,33]. Sections 2, 3 and 4 deal with univariate polynomial factorization over the previously mentioned domains. Section 5 discusses the multivariate factorization problem over these domains with emphasis on integer coefficients. We conclude with a list of open problems in section 6.

2. Factorization of univariate polynomials over finite fields

The exposition of this problem given in D. Knuth's book [22, Sec. 4.6.2] is quite complete, and we refer the reader who seeks an introduction to current algorithms to that work. We wish to mention here that testing a polynomial $f(x) \varepsilon Z_p[x]$ of degree n for irreducibility can be achieved in $O(n^2 \log(n)^3 \log(p))$ arithmetic steps using the distinct degree factorization [41]. This bound is polynomial in n log(p) which is the order of the size needed to encode f on the tape of a Turing machine. The distinct degree factorization algorithm also produces for each factor of degree m the product polynomial of all factors of degree m. Though this is still an incomplete factorization, it is a further refinement of the squarefree decomposition and no corresponding algorithm is known for integers. It is the goal of any probabilistic factorization algorithm to make the expected running time polynomial in n log(p). As we have noted in the introduction, M. Rabin's algorithm, which is based on finding the roots of f in some larger Galois field of characteristic p, achieves this goal. Recently D. Cantor and H. Zassenhaus proposed a probabilistic version of Berlekamp's algorithm which takes $O(n^3 + n^2\log(n)\log(p)^3)$ expected steps [8]. It is not clear which of the algorithms is more efficient in practice [6] though the root-finding algorithm has been proven to be asymptotically faster [4]. Further probabilistic improvements to the Berlekamp algorithm are reported in [26]. There is no known deterministic complete factorization algorithm whose worst time complexity is polynomial in n log(p), except in the case that p-1 is highly composite (e.g. $p = L\, 2^k+1$ with L of the same size as k). In that case R. Moenck devised an algorithm very similar to the root finding algorithm which factors f in $O(n^3 + n^2 \log(p) + n \log(p)^2)$ steps [29]. So far we have only addressed the problem where the coefficients lie in a prime residue field but, as one might have expected, most of the above algorithms can be modified to also work in Galois fields of order p^k, $k>1$.

3. Factorization of univariate polynomials over the integers

Given a polynomial $\bar{h}(x) \varepsilon Z[x]$ we seek to compute its content (see chapter on "Arithmetic") and all its primitive irreducible polynomial factors $g_{ij}(x) \varepsilon Z[x]$, that is

$$\bar{h}(x) = \mathrm{cont}(\bar{h})\prod_{i=1}^{r}(\prod_{j=1}^{s_i} g_{ij}(x))^i$$

with all g_{ij} irreducible and pairwise distinct. The complete algorithm consists of three separate steps, namely

[Factorization of $\bar{h}(x) \in Z[x]$]

(C) [Content computation:] The integer GCD of all coefficients of $\bar{h}$ constitutes the $\mathrm{cont}(\bar{h})$, $h \leftarrow \bar{h}/\mathrm{cont}(\bar{h})$ [h is now a primitive polynomial.]

(S) [Squarefree decomposition of h:] Compute squarefree polynomials $f_i(x) \in Z[x]$, $1 \le i \le r$, $\mathrm{GCD}(f_j,f_k) = 1$ for $1 \le j \ne k \le r$ such that $h(x) = \prod_{i=1}^{r} (f_i(x))^i$.

(F) [Factor the squarefree f_i:] FOR i=1,...,r DO

Compute irreducible polynomials $g_{ij}(x) \in Z[x]$, $1 \le j \le s_i$ such that $f_i(x) = \prod_{j=1}^{s_i} g_{ij}(x)$. □

Step (C) is a repeated integer GCD computation and shall not be discussed further.

The computational aspects of step (S) were first investigated by E. Horowitz following an idea of R. Tobey in 1969 (cf. [16]) whose algorithms were later improved by D. Musser [33], D. Yun [58] and P. Wang and B. Trager [54]. We shall briefly present D. Yun's algorithm:

[Squarefree decomposition of a primitive polynomial h:]

(S1) $g(x) \leftarrow \mathrm{GCD}(h(x), dh(x)/dx)$ where $dh(x)/dx = h'(x)$ is the derivative of h w.r.t. x.

$c_1(x) \leftarrow h(x)/g(x)$; $d_1(x) \leftarrow (dh(x)/dx)/g(x) - dc_1(x)/dx$. [Assume that $h = \prod_{i=1}^{r} f_i^i$ with the f_i squarefree and pairwise relatively prime. Then $g = \prod_{i=2}^{r} f_i^{i-1}$, $c_1 = \prod_{i=1}^{r} f_i$, $h'/g = \sum_{i=1}^{r} \left[i\, f_i' \prod_{j=1,j\ne i}^{r} f_j \right]$ which is relatively prime to g since $\mathrm{GCD}(f_i,f_i') = 1$ (The f_i are squarefree!). Thus $d_1 = \sum_{i=2}^{r} \left[(i-1) f_i' \prod_{j=1,j\ne i}^{r} f_j \right]$.]

(S2) FOR $k \leftarrow 1,2,\ldots$ UNTIL $c_k = 1$ DO

[At this point $c_k = \prod_{i=k}^{r} f_i$, $d_k = \sum_{i=k+1}^{r} \left[(i-k) f_i' \prod_{j=k,j\ne i}^{r} f_j \right]$.]

$f_k(x) \leftarrow \mathrm{GCD}(c_k(x),d_k(x))$; $c_{k+1}(x) \leftarrow c_k(x)/f_k(x)$; $d_{k+1}(x) \leftarrow d_k(x)/f_k(x) - dc_{k+1}(x)/dx$. □

The reader should be able to derive the correctness of this algorithm from the embedded comments. It is important that the cofactor of h' in the GCD computation of step (S1) and that of d_k in step (S2) are relatively prime to the computed GCDs. This enables one to use, besides the modular GCD algorithm, the EZGCD algorithm [31] whose general version needs the above algorithm if both cofactors have a common divisor with the GCD. The relation between polynomial GCDs and squarefree decompositions is even more explicit (cf. [59]).

Step (F) is the actual heart of the algorithm. As outlined in the introduction, various substeps are needed for the Berlekamp-Hensel algorithm:

[Factorization of a primitive, squarefree polynomial f:]

(F1) [Choice of a modulus:] Find a prime number p which neither divides ldcf(f(x)) nor the resultant of f(x) and df(x)/dx. The latter is equivalent to the condition that f(x) modulo p is squarefree. By trying various primes in connection with the distinct factorization procedure we may also attempt to minimize the number of modular factors in the next step.

(F2) [Modular factorization:] Factor f(x) modulo p completely, namely compute irreducible polynomials $u_1(x),...,u_r(x) \varepsilon Z_p[x]$ such that $\mathrm{ldcf}(u_1) \equiv \mathrm{ldcf}(f)$ (modulo p), $u_2,...,u_r$ are monic and $u_1(x)...u_r(x) \equiv f(x)$ (modulo p).

(F3) [Factor coefficient bound:] Compute an integer B(f) such that all coefficients of any possible factor of f(x) in Z[x] are absolutely bounded by B(f) (see chapter on ‘Useful Bounds’).

(F4) [Lift modular factors:] $q \leftarrow p$;

FOR $k \leftarrow 1,2,...$ UNTIL $q \geq 2\,B(f)$ DO

$q \leftarrow q^2$; [At this point $q = p^{2^k}$.]

Compute polynomials $u_i^{(k)}(x) \varepsilon Z_q[x]$ such that $u_1^{(k)} \cdots u_r^{(k)} \equiv f(x)$ (modulo q), $\mathrm{ldcf}(u_1^{(k)}) \equiv \mathrm{ldcf}(f)$ (modulo q) and $u_i^{(k)} \equiv u_i$ (modulo p) where the coefficients of $u_i^{(k)}$ are interpreted as p-adic approximations.

(F5) [Form trial factor combinations:]
$h(x) \leftarrow f(x)$; $C \leftarrow \{2,\ldots,r\}$; $s \leftarrow 0$; $j \leftarrow 1$;

REPEAT $t \leftarrow s$;
FOR $m \leftarrow j ,\ldots,$ cardinality of C DO
FORALL subsets $\{i_1,\ldots,i_m\}$ of C DO
Test whether $g(x) = \mathrm{pp}(\mathrm{ldcf}(h)\, u_{i_1}^{(k)} \cdots u_{i_m}^{(k)}$ (modulo p^{2^k})) divides h, where k is the number of iterations in (F4) and the modulus is balanced before taking the primitive part over the integers. If so then set $s \leftarrow s+1$; $g_s(x)$

$\leftarrow g(x)$; $h(x) \leftarrow h(x)/g(x)$; $j \leftarrow m$; $C \leftarrow C$ minus $\{i_1, \ldots, i_m\}$; and exit both FOR loops.

END FORALL

END FOR

UNTIL $t = s$ [No more factors discovered in the FOR loops]

$s \leftarrow s+1$; $g_s(x) \leftarrow h(x)$

[All factors are computed as $f(x) = g_1(x)\cdots g_s(x)$.] □

We must scrutinize various steps further. By the choice of p in step (F1) $\overline{f}(x) = f(x)$ modulo p is of the same degree as f(x) and the inverse of $\mathrm{ldcf}(\overline{f})$ exists in Z_p. We factor the monic polynomial $\mathrm{ldcf}(\overline{f})^{-1}\overline{f}(x)$ first into distinct degree factors and then into irreducibles in step (F2). To satisfy the condition on the $\mathrm{ldcf}(u_1)$ we multiply the monic u_1 by $\mathrm{ldcf}(\overline{f})$ in Z_p. Step (F4) utilizes the "Hensel-lemma" and various lifting techniques have been investigated [60], [32], [51] (see also the chapter on "Homomorphic Images"). The following algorithm is due to P. Wang:

[Hensel Lifting Algorithm:]

[Given polynomials $f(x) \in Z[x]$, q relatively prime to ldcf(f), $u_1^*(x),\ldots,u_r^*(x) \in Z_q[x]$ such that $\mathrm{ldcf}(u_1) \equiv \mathrm{ldcf}(f)$ (modulo q), $u_2^*,\ldots,u_r^*$ monic and

$$u_1^*(x)\cdots u_r^*(x) \equiv f(x) \text{ (modulo q)}. \tag{1}$$

Furthermore given polynomials $v_1^*(x),\ldots,v_r^*(x) \in Z_q[x]$ with $\deg(v_i^*) < \deg(u_i^*)$ for $1 \le i \le r$, and if we set $\tilde{u}_i^* = \prod_{j=1,j\ne i}^{r} u_i^*$ then

$$v_1^*(x)\tilde{u}_1^*(x) +\ldots+ v_r^*(x)\tilde{u}_r^*(x) \equiv 1 \text{ (modulo q)}.$$

The goal is to produce polynomials $u_1^{**}(x),\ldots,u_r^{**}(x)$, $v_1^{**}(x),\ldots,v_r^{**}(x) \in Z_{q^2}[x]$ which satisfy the same conditions as the single-starred polynomials if we replace the modulus q by q^2.]

(H1) Replace $\mathrm{ldcf}(u_1^*)$ by ldcf(f) modulo q^2;

[Lift u_i^* by computing $\hat{u}_i^* \in Z_q[x]$ such that $u_i^{**} = u_i^* + q\hat{u}_i^*$ with $\deg(\hat{u}_i^*) < \deg(u_i^*)$ for $i \ge 1$.]

$$t(x) \leftarrow \left[f(x) - \prod_{i=1}^{r} u_i^*(x) \right] \text{ modulo } q^2;$$

[The above replacement guarantees deg(t) < deg(f). Also all coefficients of t are divisible by q because of (1).]

$t(x) \leftarrow t(x)/q$; [Integer division, hence $t(x) \in Z_q[x]$. We need to determine $\hat{u}_i^*$ with

$$\hat{u}_1^*(x)\tilde{u}_1^*(x) +\ldots+ \hat{u}_r^*(x)\tilde{u}_r^*(x) = t(x). \tag{2}$$
]

FOR $i \leftarrow 1,\ldots,r$ DO

$\hat{u}_i^*(x) \leftarrow \mathrm{remainder}(t(x)v_i^*(x), u_i^*(x))$ in $Z_q[x]$; $u_i^{**}(x) \leftarrow u_i^*(x) + q\hat{u}_i^*(x)$.

[Obviously the polynomials tv_i^* solve (2) but do not satisfy the degree constraint for the $\hat{u}_i^*$. Hence the $\hat{u}_i^*$ solve (2) modulo $\prod_{i=1}^{r} u_i^*$ but since all degrees are less than deg(f) there must be equality.]

(H2) [Lift v_i^* by computing $\hat{v}_i^* \in Z_q[x]$ such that $v_i^{**} = v_i^* + q\hat{v}_i^*$ and $\deg(\hat{v}_i^*) < \deg(u_i^*)$.]

$b(x) \leftarrow \left[\left[1 - \sum_{i=1}^{r} v_i^*(x)\tilde{u}_i^*(x)\right] \text{ modulo } q^2\right] / q$; [Again the division is integral and $b(x) \in Z_q[x]$ with $\deg(b) < \deg(f)$.]

FOR $i \leftarrow 1,\ldots,r$ DO

$\hat{v}_i^*(x) \leftarrow \mathrm{remainder}(b(x)v_i^*(x), u_i^*(x))$ in $Z_q[x]$; $v_i^{**}(x) \leftarrow v_i^*(x) + q\hat{v}_i^*(x)$.□

In order to use the above algorithm within the loop of step (F4) we also need to initialize the $v_i(x)$ in Z_p with $1/\prod_{i=1}^{r} u_i(x) = \sum_{i=1}^{r} v_i(x)/u_i(x)$ and $\deg(v_i) < \deg(u_i)$. To do this one can use the extended Euclidian algorithm r-1 times (see chapter on "Remainder Sequences") or use fast partial fraction decomposition algorithms [24], [1].

Step (H2) is not necessary if one only considers the first solution v_i and corrects u_i^* from modulus q to modulus pq by calculating $\hat{u}_i^*$ in $Z_p[x]$. This method is referred to as "linear lifting" whereas our algorithm has quadratic p-adic convergence. We also lift all factors in parallel while earlier versions proceeded with one factor and its cofactor at a time. It is not clear which technique is preferable (cf. [57], [61]), though the parallel quadratic approach seems superior [51]. In order to prevent p^{2^k} from overshooting B(f) by too much one may calculate the last correction polynomials $\hat{u}_i^*$ with a smaller modulus than q.

As we will see below, in the worst case step (F5) is the dominant step in our algorithm. Therefore one is advised to test first whether the second highest coefficient is absolutely smaller than $\deg(f)\,|f|_2$, the corresponding factor coefficient bound (see chapter on 'Useful Bounds') [43], or whether the constant coefficient of g(x) divides that of f(x).

D. Musser has carefully analysed a variation of steps (F1) - (F5), the result of which is the following [32]: Let $f = g_1 \cdots g_s$ in Z[x], $\deg(g_1) \le \deg(g_2) \le \cdots \le \deg(g_s)$, and let

$$\mu = \begin{cases} \max\limits_{i=2,\ldots,s} \{\deg(g_{i-1}), \lfloor \deg(g_i)/2 \rfloor\} & \text{if } s>1 \\ \lfloor \deg(f)/2 \rfloor & \text{if f is irreducible}. \end{cases}$$

If f factors into r polynomials modulo p then

$$\min(2^r, r^\mu)\ \mu n^2\ (n+\log(B(f)))^2$$

dominates the complexity of the factorization problem. This bound depends intrinsically on r which is one reason why one should attempt to minimize this number in step (F2). If one does not, the algorithm still performs quite well − on the average. An n-th degree polynomial in $Z_p[x]$ has an average of log(n) factors as p tends to infinity and 2^r averages n+1 where r is the number of modular factors (cf. [22, Sec.4.6.2., Exercise 5]. However, almost all integer polynomials are irreducible (cf. [22, Sec.4.6.2, exercise 27]), and one may not expect almost all inputs to our algorithm to behave that way since a user probably tries to factor polynomials which are expected to be composite. In this matter G. Collins has shown, subject to two conjectures, that if we restrict our set to those polynomials which factor over the integers into factors of degree $d_1, d_2, \ldots, d_s$ for a given additive decomposition of $n = d_1+\ldots+d_s$, the average number of trial combinations will be below n^2. This result only holds if one processes combinations of m factors at a time as we did in step (F5) ("cardinality procedure"), because if one chooses to test combinations of a possible total degree ("degree procedure") the average behavior may be exponential in n [10].

The worst case complexity of the Berlekamp-Hensel algorithm is unfortunately exponential in n, the degree of f. This is because, as we will prove below, there exist irreducible integer polynomials of arbitrarily large degree which factor over every prime into linear or quadratic factors. This means that we must test at least $2^{n/2-1}-1$ trial factor combinations to show that no integral factor occurs. The following theorem is attributed to H.P.F. Swinnerton-Dyer by E. Berlekamp [5, pp.733-734].

Theorem: Let n be an integer and $p_1, \ldots, p_n$ positive distinct prime numbers. Then the monic polynomial $f_{p_1,\ldots,p_n}(x)$ of degree 2^n whose roots are $e_1\sqrt{p_1}+\ldots+e_n\sqrt{p_n}$ with $e_i=\pm1$ for $1\le i\le n$ has integral coefficients and is irreducible in Z[x]. Moreover, for any prime q, $f_{p_1,\ldots,p_n}(x)$ modulo q factors into irreducible polynomials in $Z_q[x]$ of at most degree two.

Proof: In the following we assume that the reader is familiar with some basic facts of Galois theory. The book by van der Waerden [44] is a standard reference whose notation we adopt. The following abbreviations are useful. $f_k(x)\equiv f_{p_1,\ldots,p_k}(x)$ and $K_k\equiv Q(\sqrt{p_1},\ldots,\sqrt{p_k})$ for $1\le k\le n$. By induction we prove that $f_n(x)\ \varepsilon\ Z[x]$, $[K_n:Q] = 2^n$, and that $\theta = \sqrt{p_1}+\ldots+\sqrt{p_n}$ is a primitive element of K_n. For n=1 the facts are trivial. It follows from the hypothesis $f_{n-1}(x)\ \varepsilon\ Z[x]$ and from $f_n(x) = f_{n-1}(x+\sqrt{p_n})\ f_{n-1}(x-\sqrt{p_n})$ that $f_n(x)\ \varepsilon\ Z[\sqrt{p_n}, x]$ with coefficients that are symmetric in the two conjugates $\sqrt{p_n}$ and $-\sqrt{p_n}$. By the fundamental theorem on symmetric functions the coefficients must be integers. (Actually $f_n(x) = \mathrm{res}_y(f_{n-1}(x-y), x^2-p_n)$ as is shown in the chapter on "Algebraic Domains".) From the second hypothesis, namely $[K_{n-1}:Q]=2^{n-1}$, we conclude that the set

$$B_k = \{1\} \cup \{\sqrt{p_{i_1}\cdots p_{i_j}} \mid j=1,\ldots,k,\ 1\le i_1<i_2<\cdots<i_j\le k\}$$

forms a basis for K_k over Q, $1\le k\le n-1$.

We show by induction that $\sqrt{p_n}$ does not lie in the Q-span of B_{n-1}. Assume it does, namely there exist rationals $r_0, r_{i_1,\ldots,i_j}$ such that

$$r_0 + \sum_{1\le i_1<\ldots<i_j\le n-1} r_{i_1,\ldots,i_j}\sqrt{p_{i_1}\cdots p_{i_j}} = \sqrt{p_n}. \quad (1)$$

Since p_n is a new prime, at least two coefficients on the left hand side of (1) are non-zero. Then for the two corresponding basis elements there exists a p_k such that $\sqrt{p_k}$ occurs in one but not the other. Without loss of generality assume $p_k=p_{n-1}$. Then (1) can be rewritten as

$$s_0 + s_1\sqrt{p_{n-1}} = \sqrt{p_n},\ s_0,\ s_1 \ \varepsilon\ K_{n-2}. \quad (2)$$

Since s_0 and s_1 are linear combinations in B_{n-2} with a non-zero coefficient it follows that both $s_0 \ne 0$ and $s_1 \ne 0$. Squaring (2) then leads to $\sqrt{p_{n-1}} = (p_n - s_0^2 - s_1^2 p_{n-1})/2s_0s_1 \ \varepsilon\ K_{n-2}$ in contradiction to the induction hypothesis.

Therefore $[K_n:K_{n-1}]=2$ and hence $[K_n:Q]=2^n$. We proceed to show that $K_n=Q(\theta)$. Let $\alpha_1=\sqrt{p_1}+\ldots+\sqrt{p_{n-1}}$, $\alpha_2,\ldots,\alpha_{2^{n-1}}$ be the roots of $f_{n-1}(x)$ and consider the polynomials $g_1(x) = f_{n-1}(\alpha_1+\sqrt{p_n}-x)$ and $g_2(x) = x^2-p_n$. Obviously g_1, $g_2\ \varepsilon\ Q(\theta)[x]$ and $\sqrt{p_n}$ is a common root. However, $g_1(-\sqrt{p_n})\ne 0$ because $\alpha_1+2\sqrt{p_n} \ne \alpha_i$ for $1\le i\le 2^{n-1}$ since $\alpha_i-\alpha_1\ \varepsilon\ K_{n-1}$ but $\sqrt{p_n}\ \varepsilon\!\!\!|\ K_{n-1}$. Therefore $GCD(g_1, g_2) = x-\sqrt{p_n}\ \varepsilon\ Q(\theta)[x]$ and hence $Q(\theta) = Q(\alpha_1,\sqrt{p_n})$. By hypothesis $K_{n-1} = Q(\alpha_1)$ which gives $Q(\theta)=K_n$. The irreducibility of f_n now follows quickly. The minimal polynomial of θ has degree $[Q(\theta):Q]=[K_n:Q]=2^n$ and therefore f_n is this irreducible polynomial.

The factorization property modulo q can be proven by the following argument. Since $\sqrt{p_i}$ modulo q $\varepsilon\ GF(q^2)$ for $1\le i\le n$ all roots of f_n modulo q lie in $GF(q^2)$. If f_n modulo q had an irreducible factor of degree m>2 its roots would generate the larger field $GF(q^m)$ and could not be elements of $GF(q^2)$. □

The construction of $f_{p_1,\ldots,p_n}$ has been generalized using higher radicals instead of square roots [20] and it can be shown that $\log(|f_{p_1,\ldots,p_n}|) = O(2^n\log(n))$ which makes the worst case of the Berlekamp-Hensel algorithm truly exponential in its input size. Here the following remark is in place. We always assume that our algorithm operates on densely encoded polynomials. If we allow sparse encoding schemes, various primitive operations on the input polynomials such as GCD computations are NP-hard (cf. [38, 39]) and the factorization problem actually requires exponential space. In order to substantiate the last claim we consider the polynomial x^n-1 whose sparse encoding requires O(log n) bits. However, R. Vaughan [45] has shown that for infinitely many n the cyclotomic polynomials Ψ_n, which constitute irreducible factors of x^n-1, have coefficients absolutely larger than $\exp(n^{\log 2/\log\log n})$.

One question about our algorithm remains to be answered. That is how the choice of various primes in step (F1) can influence later steps, especially step (F5). It is clear that if a polynomial f factors modulo p_1 into all quadratic and modulo p_2 into all cubic factors, then the degrees of integral factors must be multiples of 6. Indeed if the degree sets of factorizations modulo various primes are completely incompatible we know the input polynomial to be irreducible without the need of steps (F2) - (F5). For this situation D. Musser has developed an interesting model which, given a random irreducible polynomial $f(x) \in Z[x]$ of degree n, shows how to derive the average number $\mu(n)$ of factorizations modulo distinct primes $p_1,\ldots,p_{\mu(n)}$ needed to prove f irreducible [34]. His approach is based on the fact that the degrees $d_1,\ldots,d_r$ of a factorization $f \equiv g_1 \cdots g_r$ modulo p, $d_i = \deg(g_i)$ for $1 \le i \le r$ and p a random prime correspond to the cycle lengths of a random permutation $(1,...,d_1)(d_1+1,...,d_1+d_2)...(d_1+...+d_{r-1}+1,...,d_1+...+d_r)$ of n elements. The Swinnerton-Dyer polynomials $f_{p_1,\ldots,p_n}$ of the previous theorem obviously do not satisfy this property but it remains valid for any specific polynomial f provided that the Galois group of f is the full symmetric group. Our statement is somewhat stronger than what D. Musser proves because the latter follows from the fact that almost all polynomials have the symmetric group as Galois group [13]. Our claim is a consequence of the Chebotarev Density Theorem [35, Chap.8.3]. This theorem also applies to the Swinnerton-Dyer polynomials, and an effective version has been used to construct succinct certificates for normal polynomials, i.e., $N=\{f \mid f \in Z[x], \text{ f irreducible and normal}\} \in NP$ [19]. D. Cantor has recently shown the same by more elementary means for generally irreducible polynomials, i.e., $I=\{f \mid f \in Z[x], \text{ f irreducible}\} \in NP \cap co{-}NP$ [7]. However, in P. Weinberger's algebraic number theoretic proof showing that the Generalized Riemann Hypothesis implies $I \in P$, the Chebotarev Density Theorem again plays an important rôle [55]. *4. Factorization of univariate polynomials over algebraic extensions of Q*

The decidability of factoring a polynomial $f(x) \in Q(\theta)[x]$, θ an algebraic number, goes back to L. Kronecker [23, sec.4, pp. 12-13]. The same algorithm can also be found in [44, pp.136-137] which has been adopted and improved for computer usage by B. Trager [42]. However, again the Hensel lemma provides a more efficient method. We shall briefly outline the ideas involved and refer the reader to P. Weinberger's and L. Rothschild's paper [56] for the details. Without loss of generality, we may assume that θ is an algebraic integer with minimal polynomial $h(\theta) \in Z[\theta]$ of degree m (see chapter on "Algebraic Domains" for terminology.) We seek to factor $f(x) \in Q(\theta)[x]$ of degree n which we can assume to be monic. Let

$$f(x) = x^n + 1/d \sum_{i=0}^{n-1} \left[\sum_{j=0}^{m-1} b_{ij}\theta^j \right] x^i$$

with $d, b_{ij} \in Z$. It can be shown [56, Sec.8] that if $g(x) \in Q(\theta)[x]$, monic, and g(x) divides f(x), then

$$g(x) = x^k + 1/(dD) \sum_{i=0}^{k-1} \left[\sum_{j=0}^{m-1} c_{ij}\theta^j \right] x^i$$

with $k<n$, $c_{ij} \in Z$ and D the discriminant of h, $D = \mathrm{res}(h, dh/d\theta)$. Furthermore, there is an effective bound B of length polynomial in $n \log(|f|)$ such that $|c_{ij}| \le B$ for

$0 \le i \le k-1$, $0 \le j \le m-1$ [56, Sec.8]. Let p be a prime number such that p divides neither d nor D and that $\overline{h}(\theta) = h(\theta) \bmod p$ is irreducible. The last condition may not be satisfiable but we shall defer that case for later. Then the coefficients of $\overline{f}(x) = f(x)$ modulo p can be viewed as elements of $GF(p^m)$ generated by a root of $\overline{h}(\theta)$. We factor $\overline{f}(x) = \overline{g_1}(x)...\overline{g_r}(x)$ over this finite field, i.e.,

$$\overline{g_s}(x) = x^{k_s} + \sum_{i=0}^{k_s-1}\left[\sum_{j=0}^{m-1} \overline{c_{ij}}^{(s)}\theta^j\right]x^i,\ \overline{c_{ij}}^{(s)} \in Z_p\ (1 \le s \le r),$$

$k_1+...+k_r = n$. We can now lift the factors into a larger residue domain $q = p^k \ge 2B$ adjoined by a root of $\tilde{h}(\theta) = h(\theta)$ modulo q. By multiplication of dD modulo q we obtain $f(x) \equiv \tilde{g}_1(x)\cdots\tilde{g}_r(x)$ modulo $(q, \tilde{h}(\theta))$ with

$$\tilde{g}_s(x) = x^{k_s} + 1/(dD)\sum_{i=0}^{k_s-1}\left[\sum_{j=0}^{m-1} \tilde{c}_{ij}^{(s)}\theta^j\right]x^i$$

with $\tilde{c}_{ij}^{(s)}$ balanced residues in Z_q. It remains to test whether any trial combination of factors $\tilde{g}_s(x)$ constitutes an actual factor.

If $\overline{h}(\theta)$ factors for all primes, we then can perform the lifting modulo any factor $h^*(\theta)$ of $\overline{h}(\theta)$. To construct factors $\tilde{g}_s(x)$ modulo $(q, \tilde{h}(\theta))$ we may either use the Chinese remainder theorem [56, Sec.10] or the lattice algorithm by A. Lenstra [27,28]. Both algorithms are, however, of exponential complexity in the number of modular factors.

5. Factorization of multivariate polynomials

We shall begin this chapter with Kronecker's algorithm which, for certain coefficient domains (such as C), is still the only one known.

[Factorization of $f(x_1,\ldots,x_v) \in D[x_1,...,x_v]$ with D being a unique factorization domain.]

(K1) [Compute degree bound:] Obtain an integer d larger than the degree of f in any single variable.]

(K2) [Reduction:] $\overline{f}(y) \leftarrow S_d(f) = f(y, y^d, \ldots, y^{d^{v-1}})$.

(K3) [Factorization:] Factor $\overline{f}(y)$ into irreducibles, i.e., $\overline{f}(y) = \overline{g_1}(y)...\overline{g_s}(y)$, $\overline{g_i}(y) \in D[y]$ for $1 \le i \le s$.

(K4) [Inverse reduction and trial division:] For all products $\overline{g_{i_1}}(y)...\overline{g_{i_m}}(y)$ (similar to step (F5) in section 3) perform the following test:

$g_{i_1,\ldots,i_m}(x_1,\cdots,x_v) \leftarrow S_d^{-1}(\overline{g_{i_1}}\cdots\overline{g_{i_m}})$ where S_d^{-1} is the inverse of S_d which is additive and $S_d^{-1}(\lambda y^{b_1+db_2+\ldots+d^{v-1}b_v}) = \lambda x_1^{b_1}\cdots x_v^{b_v}$ with $0 \le b_i < d$ for $1 \le i \le v$, $\lambda \in Z$.

Test whether $g_{i_1,\ldots,i_m}$ divides f and if so remove this irreducible factor from f and proceed with its co-factor. □

The correctness of this algorithm follows easily from the fact that no variable in any factor of f can occur with degree d or higher. The running time of the algorithm depends on of how fast the univariate polynomial $f(\overline{y})$ can be factored, the degree of which can be substantially large. It should be clear that step (K4) can take time exponential in the degree of f, e.g., if D = C and f is irreducible. Unfortunately this exponential worst case complexity remains true for D = Z [19]. In this case, the Hensel lemma has produced a much more efficient approach. In the following we will take a closer look at this algorithm.

The overall structure of the multivariate factorization algorithm is remarkably close to that of the univariate algorithm of section 3. First we choose a main variable x, i.e., the input polynomial $\overline{h}\ \varepsilon\ Z[y_1,\ldots,y_v,x]$. The content computation of step (C) now becomes a GCD computation in $Z[y_1,\ldots,y_v]$. The squarefree decomposition performed in step (S) can also be achieved by D. Yun's algorithm if we replace the derivatives d/dx by partial derivatives $\partial/\partial x$ and the GCDs by multivariate polynomial GCDs. However, in this case P. Wang's and B. Trager's algorithm becomes more efficient [54].

The idea of their algorithm is to find an evaluation point $(b_1,\ldots,b_v)$ such that if $h(y_1,\ldots,y_v,x) = \prod_{i=1}^{r} f_i(y_1,\ldots,y_v,x)^i$ is the squarefree decomposition of h, and $h(b_1,\ldots,b_v,x) = \hat{h}(x) = \prod_{i=1}^{\hat{r}} \hat{f}_i(x)^i$ is that of $\hat{h}$, then $r = \hat{r}$ and $f_i(b_1,\ldots,b_v,x) = \hat{f}_i(x)$, $1 \le i \le r$. Under these conditions f_r divides $g = 1/(r-1)!(\partial/\partial x)^{r-1}(h)$, $\hat{f}_r$ divides $\hat{g} = 1/(r-1)!(d/dx)^{r-1}\hat{h}(x)$ and we can lift the equation

$$g(y_1,\ldots,y_v,x) \equiv \hat{f}_r(x)(\hat{g}(x)/\hat{f}_r(x)) \text{ modulo } (y_1-b_1,\ldots,y_v-b_v)$$

to determine f_r from the univariate square decomposition of $\hat{h}$, provided $\hat{g}/\hat{f}_r \ne 1$. Evaluation points for which the above conditions do not hold are, as in the modular multivariate GCD algorithm, very rare.

Step (F), the complete factorization of a squarefree polynomial $f(y_1,\ldots,y_v,x)$, is again a major challenge. As in the above squarefree decomposition algorithm we evaluate the minor variables y_i at integers $b_i, 1 \le i \le v$ then factor the resulting univariate polynomial $f(b_1,\ldots,b_v,x)$ and finally rebuild multivariate factor candidates by a Hensel lifting algorithm with respect to the prime ideal p generated by $\{(y_1-b_1),\ldots,(y_v-b_v)\}$. Instead of presenting a complete algorithm we shall work out a simple example and refer the reader to the papers by P. Wang [53,49,50,52] and D. Musser [33] for the details.

Example: Factor

$f(y,z,x) = x^3+((y+2)z+2y+1)x^2$

$+((y+2)z^2+(y^2+2y+1)z+2y^2+y)x$

$+(y+1)z^3+(y+1)z^2+(y^3+y^2)z+y^3+y^2.$

The polynomial is monic and squarefree.

Step F1: Choose an evaluation point which preserves degree and squarefreeness but contains as many zero components as possible.

$y=0$, $z=0$: $f(0,0,x) = x^3 + x^2$ is not squarefree

$y=1$, $z=0$: $f(1,0,x) = x^3 + 3x^2 + 3x + 2$ is squarefree.

Translate variables for nonzero components

$$f(w+1,z,x) = x^3+3x^2+3x+2+(2x^2+5x+5)w$$
$$+(2x+4)w^2+w^3+((3x^2+4x+2)+(x^2+4x+5)w$$
$$+(x+4)w^2+w^3)z+((3x+2)+(x+1)w)z^2+(2+w)z^3$$

By $f_{ij}(x)$ we denote the coefficient of $w^j z^i$.

Step F2: Factor $f_{00}(x) = g_{00}(x)h_{00}(x)$ in $Z[x]$. We get $x^3 + 3x^2 + 3x + 2 = (x+2)\ (x^2 + x + 1)$.

Step F3: Compute highest degrees of w and z in factors of $f(w+1,z,x) = g(w,z,x)h(w,z,x)$: $\deg_w(g,h) \le 3$, $\deg_z(g,h) \le 2$.

Step F4: Lift g_{00} and h_{00} to highest degrees in w and z. We set $g(w,z,x) = g_{00}(x) + g_{01}(x)w + g_{02}(x)w^2 + \cdots + (g_{10}(x) + g_{11}(x)w + ...)z + \cdots$ and $h(w,z,x) = h_{00}(x) + h_{01}(x)w + h_{02}(x)w^2 + \cdots + (h_{10}(x) + h_{11}(x)w + ...)z + (h_{20}(x) + h_{21}(x)w + ...)z^2 + \cdots$ and compute $g_{01},h_{01},g_{02},h_{02},...,g_{10},h_{10},g_{11},h_{11},...,g_{20},h_{20},\cdots$ in that sequence. Since f is monic $\deg(g_{ij}) \le 1$ and $\deg(h_{ij}) \le 2$ for $i+j \ge 1$. Multiplying g times h with undetermined g_{ij}, h_{ij} we get $g_{00}h_{01} + h_{00}g_{01} = f_{01}$ whose unique solution is $(x+2)(x+2) + (x^2+x+1).1 = 2x^2+5x+5$ by the extended Euclidean algorithm. In the next step we get $g_{00}h_{02} + h_{00}g_{02} = f_{02} - g_{01}h_{01}$ which is solved by $(x+2).1 + (x^2+x+1).0 = 2x+4 - 1.(x+2)$. Finally $g_{00}h_{03} + h_{00}g_{03} = f_{03} - g_{01}h_{02} - g_{02}h_{01}$, or $(x+2).0 + (x^2+x+1).0 = 1 - 1.1 - 0.(x+2)$. This gives factor candidates for $f(w+1,0,x) = ((x+2)+1.w+0.w^2)((x^2+x+1)+(x+2)w+w^2)$ and a trial division shows them to be true factors.

We now lift z: $g_{00}h_{10} + h_{00}g_{10} = f_{10}$, or $(x+2)x + (x^2+x+1).2 = 3x^2+4x+2$; $g_{00}h_{11} + h_{00}g_{11} = f_{11} - g_{01}h_{10} - g_{10}h_{01}$, or $(x+2).0 + (x^2+x+1).1 = x^2+4x+5 - 1.x - 2(x+2)$; $g_{00}h_{20} + h_{00}g_{20} = f_{20} - g_{10}h_{10}$, or $(x+2).1 + (x^2+x+1).0 = 3x+2-2x$. All other equations have 0 as their right-hand sides.

The factor candidates are $f(w+1,z,x) = \left[(x+2)+w+(2+w)z\right] \left[(x^2+x+1)+(x+2)w+w^2+xz+z^2\right]$ which are the actual factors. Setting w=y-1 we obtain $f(y,z,x) = \left[x+yz+y+z+1\right] \left[x^2+(y+z)x+y^2+z^2\right]$.

In factoring the above sample polynomial we followed the algorithm by P. Wang [50]. Our construction is actually a linear lifting technique. There is also the possibility of quadratic lifting [33], but in the multivariate case, the linear algorithm seems to be more efficient [57]. If more than two univariate factors are present, one can again lift each one iteratively or lift them in parallel as we demonstrated for the univariate case.

Various complications have been identified with the multivariate Hensel algorithm.

a) *The leading coefficient problem:* In our example we dealt with a monic polynomial in which case the leading coefficients of all factors are known. If a polynomial leading coefficient is present, one can impose it on one factor as in the univariate case, but this leads most likely to dense factor candidates. P. Wang describes an algorithm to predetermine the actual leading coefficients of the factors, which avoids this intermediate expression growth [50, Sec.3].

b) *The "bad zero" problem:* In our example, y had to be evaluated at 1 in order to preserve squarefreeness. The change of variables $y_i = w_i+b_i$ for $b_i \neq 0$ can make the polynomial $f(w_1+b_1,\ldots,w_v+b_v,x)$ dense. P. Wang suggests to compute the coefficients $f_{i_1 \cdots i_v}(x)$ of $w_1^{i_1} \cdots w_v^{i_v}$ by Taylor's formula without performing the substitution

$$f_{i_1 \cdots i_v}(x) = \frac{1}{i_1! \ldots i_v!} \left[\frac{\partial}{\partial y_1}\right]^{i_1} \cdots \left[\frac{\partial}{\partial y_v}\right]^{i_v} f(y_1,\ldots,y_v,x) \mid_{y_i=-b_i}$$

See also R. Zipple's work on preserving sparseness [64,65].

c) *The "extraneous factors" problem:* This problem is the same as in the univariate case, namely that $f(b_1,\ldots,b_v,x)$ has more factors than $f(y_1,\ldots,y_v,x)$ (in which case we call $b_1,\ldots,b_v$ "unlucky"). One immediate consequence may be that the correction coefficients $g_{i_1 \cdots i_v}(x)$, $h_{i_1 \cdots i_v}(x)$ have non-integral coefficients. In order to avoid working with denominators one can choose to work with coefficients modulo a prime which preserves the squarefreeness of $f(b_1,\ldots,b_v,x)$, and as a first step lift the coefficients. A good factor coefficient bound can be found in [14, pp.135-139]. G. Viry has employed the initial transformation $\tilde{f}(y_1,\ldots,y_v,x) = f(y_1+b_1x,\ldots,y_v+b_vx,x)$ with $\tilde{f}(0,\ldots,0,x)$ squarefree. Then $\tilde{f}(y_1,\ldots,y_v,x)$ is "normalized", meaning that if $\tilde{f}(y_1,\ldots,y_v,x) = x^n+a_1(y_1,\ldots,y_v)x^{n-1}+\ldots+a_n(y_1,\ldots,y_v)$ with $a_i \in Q[y_1,\ldots,y_n]$ then the total degrees of the coefficients satisfy $\deg_{y_1,\ldots,y_v}(a_i) \leq i$ for $1 \leq i \leq n$. Using a polyhedron representation of polynomials introduced by A. Ostrowski [37] G. Viry then shows that a factor candidate with integral coefficients derived from the lifted factorization of

$\tilde{f}(0,...,0,x)$ divides $\tilde{f}(y_1,\ldots,y_v,x)$ if and only if it is normalized. Thus the trial division can be replaced by a check for being normalized [46,47].

Various implementation issues can be found in [30]. A good set of polynomials for benchmarking an actual implementation of the factorization algorithm can be found in [9].

Little is known about the average computing time of the multivariate Hensel algorithm. The worst case complexity can be exponential in the degree of the main variable depending on what evaluation points one chooses. However, unlike in the univariate case, it cannot happen that an irreducible polynomial factors for all possible evaluations. Actually, quite the opposite is true due to the Hilbert Irreducibility Theorem.

Theorem: Let $f(y_1,\ldots,y_v,x_1,...,x_t)$ be irreducible in $Z[y_1,\ldots,y_v,x_1,\ldots,x_t]$. By $U(N)$ we denote the number of v-tuples $(b_1,\cdots,b_v) \varepsilon Z^v$ such that $|b_i| \leq N$ for $1 \leq i \leq v$ and $f(b_1,...,b_v,x_1,\ldots,x_t)$ is reducible in $Z[x_1,...,x_t]$. Then there exist constants α and C (depending on f) such that $U(N) \leq C(2N+1)^{v-\alpha}$ and $0< \alpha <1$ (cf. [25, Chap.8]).

Unfortunately, no polynomial upper bounds on the length of C seem to be known which would make the theorem useful for "realistic" evaluations. In practice lucky evaluations seem quite frequent.

However, the situation for the theoretical study of the multivariate factorization does not appear completely hopeless. It can be shown, for instance, that if $f(x_1,y_2,...,y_v,x_2) \varepsilon Z[x_1,y_2,\ldots,y_v,x_2]$, v arbitrary but fixed, is irreducible then one can compute integers $b_1,\ldots,b_v$, $c_2,\ldots,c_v$ in time polynomial in $\deg(f)\log(|f|)$ such that $f(x_1-b_1,c_2(x_1-b_2),...,c_v(x_1-b_v),x_2) \varepsilon Z[x_1,x_2]$ is also irreducible. This theorem can be extended to show that factoring multivariate polynomials with a fixed number of variables is only polynomially harder than factoring bivariate polynomials [18,19].

We will not discuss special algorithms for coefficient domains other than the integers here. Factoring polynomials in $GF(q)[y,x]$ is very similar to factoring univariate polynomials over the integers. More general algorithms can be found in [11]. A multivariate Hensel algorithm for factoring polynomials in $Q(\theta)[x_1,\ldots,x_v]$ can be found in [48]. A special problem is to test a polynomial $f(x_1,\ldots,x_v) \varepsilon Z[x_1,\ldots,x_v]$ for absolute irreducibility, that is, to test $f(x_1,\ldots,x_v)$ for irreducibility in $C[x_1,\ldots,x_v]$. The first criterion probably goes back to E. Noether [36] which also implies that if $f(x_1,...,x_r)$ is absolutely irreducible, then $f(x_1,\ldots,x_r)$ remains irreducible modulo almost all prime numbers. Unfortunately, the first such prime number may be very large. A more efficient test for absolute irreducibility can be found in [15].

6. Conclusion

We have tried to capture the current situation for the problem of factoring polynomials. We believe that various algorithms presented here will be significantly improved in the future but we also believe that some of these ideas will persist through new developments. Following is a list of open problems which the author believes should receive attention.

1. Probabilistic univariate irreducibility test: Does there exist a probabilistic algorithm which tests $f(x) \in Z[x]$ for irreducibility in expected time polynomial in $\deg(f)\log(|f|)$? Can the algorithm also find factors?

2. Deterministic univariate factorization: Does there exist an algorithm which factors $f(x) \in Z[x]$ in time polynomial in $\deg(f)\log(|f|)$? (Cf. [62])

3. Polynomial reduction from bivariate to univariate factorization: Assuming that problem 2 has a positive answer, does there exist an algorithm which factors $f(y,x) \in Z[y,x]$ in time polynomial in $\deg(f)\log(|f|)$? (Cf. [19])

4. Bivariate factorization over finite fields: Can one factor $f(y,x) \in Z_p[y,x]$ in time polynomial in $p \deg(f)$? Can one test $f(y,x)$ for irreducibility in time polynomial in $\log(p)\deg(f)$?

5. Factorization of normal univariate polynomials: Given $f(x) \in Z[x]$ irreducible and normal. Can one factor f over its own splitting field in time polynomial in $\deg(f)\log(|f|)$? Notice that a solution of problem 2 provides one for this problem.

6. Analysis of multivariate Hensel algorithm: Provide an effective version of the Hilbert irreducibility theorem. What is the average number of evaluation points needed to prove irreducibility or achieve a "lucky" evaluation?

Note added in proof: A. Lenstra, H. Lenstra, and L. Lovasz have recently solved the open problem 2. Their algorithm takes $O(\deg(f)^{12}+\deg(f)^9\log(|f|_2)^3)$ steps. The author has recently solved the open problem 3 using ideas from [62].

Acknowledgement: The author wishes to acknowledge the support he received from Prof. B. Caviness and Prof. P. Wang while writing this paper.

References

[1] Abdali, S. K., Caviness, B. F., Pridor, A. : Modular Polynomial Arithmetic in Partial Fraction Decomposition. MACSYMA *1977,* 253-261.

[2] Adleman, L. M. : On Distinguishing Prime Numbers from Composite Numbers. Proc. 21st Symp. Foundations Comp. Sci. IEEE, 387-406 (1980).

[3] Adleman, L. M., Odlyzko, A. M. : Irreducibility Testing and Factorization of Polynomials. Proc. 22nd Symp. Foundations Comp. Sci. IEEE, 409-418 (1981).

[4] Ben-Or, M. : Probabilistic Algorithms in Finite Fields. Proc. 22nd Symp. Foundations Comp. Sci. IEEE, 394-398 (1981).

[5] Berlekamp, E. R. : Factoring Polynomials over Large Finite Fields. Math. Comp. *24,* 713-735 (1970).

[6] Calmet, J., Loos, R. : Deterministic Versus Probabilistic Factorization of Integral Polynomials. EUROCAM *1982,* to appear.

[7] Cantor, D. G. : Irreducible Polynomials with Integral Coefficients Have Succinct Certificates. J. of Algorithms *2,* 385-392 (1981).

[8] Cantor, D. G., Zassenhaus, H. : On Algorithms for Factoring Polynomials over Finite Fields. Math. Comp. *36,* 587-592 (1981).

[9] Claybrook, B. G. : Factorization of Multivariate Polynomials over the Integers. ACM SIGSAM Bulletin *10,* 13 (Feb. 1976).

[10] Collins, G. E. : Factoring Univariate Polynomials in Polynomial Average Time. EUROSAM *1979,* 317-329.

[11] Davenport, J. H., Trager, B. M. : Factorization over Finitely Generated Fields. SYMSAC *1981,* 200-205.

[12] Eisenstein, F. G. : Über die Irreductibilität und einige andere Eigenschaften der Gleichung, von welcher die Theilung der ganzen Lemniscate abhängt. J. f. d. reine u. angew. Math. *39,* 160-179 (1850).

[13] Gallagher, P. X. : Probabilistic Galois Theory. AMS Proc. Symp. in Pure Math., Analytic Number Theory, 91-102 (1972).

[14] Gelfond, A. O. : Transcendental and Algebraic Numbers. New York: Dover Publ. 1960.

[15] Heintz, J., Sieveking, M. : Absolute Primality of Polynomials is Decidable in Random Polynomial Time in the Number of Variables. Springer Lecture Notes Comp. Sci. 115, 16-28 (1981).

[16] Horowitz, E. : Algorithms for Partial Fraction Decomposition and Rational Function Integration. SYMSAM *1971,* 441-457.

[17] Johnson, S. C. : Tricks for Improving Kronecker's Method. Bell Laboratories Report 1966.

[18] Kaltofen, E. : A Polynomial Reduction from Multivariate to Bivariate Integer Polynomial Factorization. ACM Proc. Symp on Theory Comp. 1982, 261-266.

[19] Kaltofen, E. : On the Complexity of Factoring Polynomials with Integer Coefficients. Ph.D. thesis, RPI (1982), in preparation.

[20] Kaltofen, E., Musser, D. R., Saunders, B. D. : A Generalized Class of Polynomials that Are Hard to Factor. SYMSAC *1981*

[21] Knuth, D. E. : The Art of Computer Propramming, vol.2, Seminumerical Algorithms. Reading, MA: Addison Wesley 1969.

[22] Knuth, D. E. : The Art of Computer Programming, vol.2, Seminumerical Algorithms, 2nd ed. Reading, MA: Addison Wesley 1981.

[23] Kronecker, L. : Grundzüge einer arithmetischen Theorie der algebraischen Grössen. J. f. d. reine u. angew. Math. *92*, 1-122 (1882).

[24] Kung, H. T., Tong, D. M. : Fast Algorithms for Partial Fraction Decomposition. SIAM J. Comp. *6*, 582-593 (1977).

[25] Lang, S. : Diophantine Geometry. New York: Interscience Publ. 1962.

[26] Lazard, D. : On Polynomial Factorization. EUROCAM *1982*, to appear.

[27] Lenstra, A. K. : Lattices and Factorization of Polynomials. ACM SIGSAM Bulletin *15*, 15-16 (Aug. 1981).

[28] Lenstra, A. K. : Lattices and Factorization of Polynomials. EUROCAM *1982*, to appear.

[29] Moenck, R. T. : On the Efficiency of Algorithms for Polynomial Factoring. Math. Comp. *31*, 235-250 (1977).

[30] Moore, P. M. A., Norman, A. C. : Implementing a Polynomial Factorization Problem. SYMSAC *1981*, 109-116.

[31] Moses, J., Yun, D. Y. Y. : The EZGCD Algorithm. Proc. 1973 ACM National Conf., 159-166.

[32] Musser, D. R. : Algorithms for Polynomial Factorization. Ph.D. thesis and TR#134, Univ. of Wisconsin 1971.

[33] Musser, D. R. : Multivariate Polynomial Factorization. J. ACM *22*, 291-308 (1976).

[34] Musser, D. R. : On the Efficiency of a Polynomial Irreducibility Test. J. ACM *25*, 271-282 (1978).

[35] Narkiewicz, W. : Elementary and Analytic Theory of Algebraic Numbers. Warsaw: Polish Science Publ. 1974.

[36] Noether, E. : Ein algebraisches Kriterium für absolute Irreduzibilität. Math. Ann. *85*, 26-33 (1922).

[37] Ostrowski, A. M. : On Multiplication and Factorization of Polynomials I, Aequationes Math. *13*, 201-228 (1975).

[38] Plaisted, D. A. : Some Polynomial and Integer Divisibility Problems are NP-Hard. SIAM J. Comp. *7*, 458-464 (1978).

[39] Plaisted, D. A. : The Application of Multivariate Polynomials to Inference Rules and Partial Tests for Unsatisfiability. SIAM J. Comp. *9*, 698-705 (1980).

[40] Pratt, V. R. : Every Prime Has a Succinct Certificate. SIAM J. Comp. *4*, 214-220 (1975).

[41] Rabin, M. O. : Probabilistic Algorithms in Finite Fields. SIAM J. Comp. *9*, 273-280 (1980).

[42] Trager, B. M. : Algebraic Factoring and Rational Function Integration. SYMSAC *1976*, 219-226.

[43] Trotter, H. F. : Algebraic Numbers and Polynomial Factorization. AMS Short Course Series, Ann Arbor 1980.

[44] van der Waerden, B. L. : Modern Algebra, vol.1. Engl. transl. by F. Blum. New York: Ungar Publ. 1953.

[45] Vaughan, R. C. : Bounds for the Coefficients of Cyclotomic Polynomials. Michigan Math. J. *21*, 289-295 (1975).

[46] Viry, G. : Factorisation des Polynômes a Plusieurs Variables a Coefficient Entiers. RAIRO Informatique Theorique *12*, 305-318 (1978).

[47] Viry, G. : Factorisation des Polynômes a Plusieurs Variables. RAIRO Informatique Theorique *14*, 209-223 (1980).

[48] Wang, P. S. : Factoring Multivariate Polynomials over Algebraic Number Fields. Math. Comp. *30*, 324-336 (1976).

[49] Wang, P. S. : Preserving Sparseness in Multivariate Polynomial Factorization. MACSYMA *1977*, 55-61.

[50] Wang, P. S. : An Improved Multivariate Polynomial Factoring Algorithm. Math. Comp. *32*, 1215-1231 (1978).

[51] Wang, P. S. : Parallel p-adic Constructions in the Univariate Polynomial Factoring Algorithm. MACSYMA *1979*, 310-318.

[52] Wang, P. S. : Analysis of the p-adic Construction of Multivariate Correction Coeeficients in Polynomial Factorization: Iteration vs. Recursion. EUROSAM *1979*, 291-300.

[53] Wang, P. S., Rothschild, L. P. : Factoring Multivariate Polynomials over the Integers. Math. Comp. *29*, 935-950 (1975).

[54] Wang, P. S., Trager, B. M. : New Algorithms for Polynomial Square-free Decomposition over the Integers. SIAM J. Comp. *8,* 300-305 (1979).

[55] Weinberger, P. J. : Finding the Number of Factors of a Polynomial. 1981, submitted.

[56] Weinberger, P. J., Rothschild, L. P. : Factoring Polynomials over Algebraic Number Fields. ACM Trans. Math. Software *2,* 335-350 (1976).

[57] Yun, D. Y. Y. : Hensel Meets Newton - Algebraic Construction in an Analytic setting. Analytic Computational Complexity. Traub J. ed. NY: Academic Press 1976.

[58] Yun, D. Y. Y. : On Squarefree Decomposition Algorithms. SYMSAC *1976,* 26-35.

[59] Yun, D. Y. Y. : On the Equivalence of Polynomial GCD and Squarefree Factorization Problems. MACSYMA *1977,* 65-70.

[60] Zassenhaus, H. : On Hensel Factorization I. J. Number Theory *1,* 291-311 (1969).

[61] Zassenhaus, H. : A Remark on the Hensel Factorization Method. Math. Comp. *32,* 287-292 (1978).

[62] Zassenhaus, H. : Polynomial Time Factoring of Integral Polynomials. ACM SIGSAM Bulletin *15,* 6-7, (May 1981).

[63] Zippel, R. E. : Probabilistic Algorithms for Sparse Polynomials. EUROSAM *1979,* 216-226.

[64] Zippel, R. E. : Probabilistic Algorithms for Sparse Polynomials. Ph.D. thesis, MIT 1979.

[65] Zippel, R. E. : Newton’s Iteration and the Sparse Hensel Algorithm. SYMSAC *1981,* 68-72.