Proc. Latin '92, I. Simon (Ed.), Springer Lect. Notes Comput. Sci., vol. 583, pp. 294–313 (1992).

Polynomial Factorization 1987–1991*

Erich Kaltofen

Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12189-3590, USA
Inter-net: `kaltofen@cs.rpi.edu`

1 Introduction

Algorithms invented in the past 25 years make it possible on a computer to efficiently factor a polynomial in one, several, or many variables with coefficients from a certain field, such as a finite field or the rational, real, or complex numbers. I have surveyed work up to 1986 in the papers (Kaltofen 1982 and 1990a). This article discusses important developments of the past five years; I also take a fresh perspective of some older results. Although a conscientious effort has been made to cover (at least by citation) the significant contributions of that period, omissions are likely, which I ask to be kindly brought to my attention.

Three parameters partition the factorization problem: first, the mathematical nature and computational representation of the coefficient domains of the input polynomial, second, that of the irreducible factors, and, third, the representation of the input polynomial and the sought irreducible factors, which depends not only on the degree and number of variables but also on properties such as sparsity. Say, for instance, that a bivariate polynomial with rational coefficients is to be factored into irreducible polynomials with real coefficients. The input polynomial as well as the factors may be represented by lists of monomials, that is terms and their corresponding non-zero coefficients. For the rational input the coefficients can be just fractions of two long integers, but the representation of the real coefficients for the factors is less standardized. One choice represents a real algebraic number by its rational minimum polynomial and an isolating interval with rational boundaries (Collins 1975), while another uses a rational linear relation of powers of a complex algebraic number that is universal for all coefficients of a single factor (Kaltofen 1990b).

The organization of this survey is governed by these distinguishing problem specifications. We first discuss the "classical univariate problems" of factoring a polynomial

*This material is based on work supported in part by the National Science Foundation under Grant No. CCR-90-06077 and under Grant No. CDA-88-05910.

represented *densely* by an array of coefficients in a finite field (§2), in the rational numbers (§3.1), and in an algebraic, Kronecker-style, extension (§3.2). We then discuss factor coefficient fields whose elements can be represented by approximations, such as complex or p-adic numbers (§3.3). The former is the classical problem of finding high precision rational approximations to the complex roots of a rational, say, polynomial. There is a well-defined notion of *sparsity* (Plaisted 1977) and *straight-line program* size (see Strassen 1990) for univariate polynomials, both of which allow for degrees being exponential in the representation size. Under these models, a few exponential lower bounds for the computational complexity of factorization can be established. For instance, some complex (reducible) factors of the cyclotomic polynomial $x^L + 1$, with $L = 2^l$, require $\Omega(\sqrt{L/l})$ complex arithmetic operations for evaluation at a complex number for x (Lipton and Stockmeyer 1976; more references can be found in Strassen 1990). Note that in the sparse model $x^L + 1$ is represented by $O(l)$ bits, the binary representation-size for the exponent L, which is also the number of arithmetic operations needed to evaluate that polynomial by repeated squaring. In §7 we will pose an open problem in this setting.

When solving a system of linear equations over a field, it is possible, by Gaussian elimination, to compute the solution by the arithmetic operations $+$, $-$, $\times$, $\div$, and by testing whether elements are equal to zero, without needing any information about the nature and representation of the coefficient field. In computer science terms, the coefficient field can be viewed as an abstract data type. It is even possible to estimate the *bit-size* of each intermediately computed field element relative to the dimension of the system and the bit-size of the input elements by observing that any such element is a fraction of two minors of the augmented input matrix (Gantmacher 1959, §2). Although much of the results in the past decade make univariate polynomial factorization over concrete coefficients fields such as the rational numbers a polynomial time process, it is long known that there are fields whose arithmetic is effective, but over which polynomial factorization becomes undecidable (van der Waerden 1930; see also Fröhlich and Shepherdson 1955).

Fortunately, the multivariate polynomial factorization problem is not any harder. It is possible to construct for fields of characteristic zero a "generic" polynomial-time algorithm that reduces the problem to factoring a univariate polynomial (§4.1). Another such generic algorithm computes the multivariate factorization into *absolutely irreducible* polynomials, which have coefficients in the algebraic closure of the input coefficient field and which do not factor in any further (transcendental) field extension (§4.2). For example, if the input polynomial has rational coefficients, the absolutely irreducible factors remain irreducible over the complex numbers; e.g., the rational bivariate polynomial $x^3 - y^2$ is absolutely irreducible. Similar examples, also with algebraic coefficients, can be constructed using Eisenstein's criterion applied to the unique factorization domain $\mathbb{C}[y]$ as coefficient domain, or Dumas's far-reaching generalization (van der Waerden 1953, §24). Such examples are useful for benchmarking implementations of algorithms.

The coefficients of the absolutely irreducible factors lie in an algebraic extension of the coefficient field of the input polynomial. One thus has to design representations for elements that are algebraic over an abstract field. In §4.2 we give an account of models that have been developed. Surprisingly, the question of testing a multivariate polynomial for absolute irreducibility can be settled by arithmetic in the coefficient field itself, that is, absolute irreducibility is a purely rational problem, like linear system solving, with no restriction on the characteristic of the coefficient field. Noether (1922) proved this fact via the construction of so-called *irreducibility forms*. Our generic polynomial-time algorithm leads to irreducibility forms of small size (§4.2), which in turn can be used to establish several effective versions of irreducibility theorems, such as the Hilbert irreducibility theorem (§5).

For polynomials with many variables one has a notion of sparsity and straight-line complexity that lies between the dense and sparse or straight-line representations discussed above for univariate polynomials. Consider polynomials with n variables of total degree d: there are $\binom{n+d}{d} \geq 2^{\min\{n,d\}}$ many possible terms, but such polynomials may have much fewer non-zero monomials, such as the Newton polynomial $x_1^d + x_2^d + \cdots + x_n^d$, or they may be given by polynomial-sized straight-line programs, such as the determinant of a $d \times d$ matrix with the entries being linear sums of $x_1, \ldots, x_n$, which can be computed division-free in at least $O(d^4 \log d)$ arithmetic operations (Berkowitz 1984). The point is that such polynomials can have degrees that are bounded by a polynomial in the size of their representations, while their dense encodings still require exponential size. Several efficient randomized algorithms have been constructed that can handle the factorization of polynomials that are input in such ways (§5).

All general purpose computer algebra systems, such as Sac-2, Derive, Macsyma, Maple, Mathematica, Reduce, and Scratchpad (marketed by The Numerical Algorithms Group Ltd. under the name Axiom) have facilities for factoring polynomials. Moreover, several implementations exist outside these systems, such as factorization algorithms over finite fields used in coding theory. These implementations are employed in diverse problem solving situations such as in Gröbner basis methodology, geometric modelling, or in the computation of closed form solutions to elliptic integrals. These implementations and applications are discussed in some more detail in §6.

Despite these dramatic theoretical and practical advances, several challenging problems, some of key importance to the symbolic-computation practitioner, remain unsolved. I will propose several such problems in §7.

2 Univariate Polynomials over Finite Fields

Striking advances on polynomial-time deterministic and fast randomized factorization over finite fields have been achieved within the past five years. The table below summarizes several classical and new algorithms and their associated (expected, if randomized) running times when factoring a univariate polynomial of degree n over a finite field with $q = p^k$ elements, which we shall denote by $\mathbb{F}_q$. If the running time is given as a function $T(n, q)$ the meaning is that the corresponding algorithm needs $O(T(n, q))$ *arithmetic operations* in $\mathbb{F}_q$ to fully factor any such input polynomial. Several algorithms call linear algebra procedures, such as null space construction of an $n \times n$ matrix, or polynomial arithmetic, such as greatest common divisor computation of two n-degree polynomials. If linear algebra is required, we write n^ω for the arising asymptotic running time, where ω is the matrix multiplication exponent — $\omega = 3$ classically, and $\omega = 2.3755$ the theoretically best (Coppersmith and Winograd 1990); fast polynomial arithmetic enters in terms of poly-log factors $(\log n)^{O(1)}$ with reasonable exponent, 2 or 3, say. The exponents of most deterministic algorithms, when given as $O(1)$, seem unfortunately not to be so reasonable.

Work prior to 1983 is described in detail in the books by Knuth (1981) and Lidl and Niederreiter (1983). In the past decade much attention has been spent, unfortunately, on controlling randomization rather than the actual computational complexity. Bach and

Author(s)	**Running time**
Tonelli (1891)	$\log p$
$k = 1$ and $n = 2$: requires a quadratic non-residue $\bmod p$; such a residue can be determined in $(\log p)^{O(1)}$ time by linear search if one assumes the validity of an extension of the Riemann hypothesis (ERH) (Ankeny 1952). Generalized to higher roots by Adleman et al. (1977).	
Arwin (1918)	$(n^2 \log q)(\log n)^{O(1)}$
Splits factors of different degree only, i.e., outputs the products of irreducible factors of the same degree; denoted by "distinct degree factorization" (see also Knuth 1981, §4.6.2). Running time improved in von zur Gathen and Shoup (1991, see below).	
Butler (1954)	$n^{\omega} + (\log q)\, n\, (\log n)^{O(1)}$
Irreducibility test based on "Q-matrix" construction also used by Berlekamp; see also Schwarz (1956).	
Berlekamp (1967)	$n^{\omega} + qn^2(\log n)^{O(1)}$
For large p the asymptotic time can be reduced by re-arranging the last phase of the algorithm.	
Berlekamp (1970)	$n^{\omega} + (\log q)\, n\, (\log n)^{O(1)}$
Randomized solution attributed to Collins and Knuth (1967) and Zassenhaus (1969, §3, last paragraph); probability of success not completely analyzed. Using a sparse linear system solver (Wiedemann 1986) this method can be run in time $(\log q)n^2(\log n)^{O(1)}$ or $(n^3 + n\log q)(\log n)^{O(1)}$ and, simultaneously, $O(n)$ space using a modification by Cantor and Zassenhaus (1981, §4).	
Moenck (1977)	$n^2(\log p)\,(\log n)^{O(1)}$
$k = 1$: $p - 1$ must be "smooth," e.g., $p = s2^t + 1$, $s = O(t)$; in addition, the method requires a primitive root for $\mathbb{F}_p$. See also von zur Gathen (1987) and Shoup (1991a); the latter reference gives a method whose running time is linear in the square root of the largest prime factor of $p - 1$. Further citations of other "special prime" results are found there as well.	
Rabin (1980)	$(\log q)\, n^3(\log n)^{O(1)}$
Randomized method based on root finding in algebraic extensions with a clever failure probability analysis; a randomized algorithm for construction irreducible polynomials of degree n over $\mathbb{F}_q$ is also presented. Improvements are found in Ben-Or (1981); see also Camion (1982).	
Cantor and Zassenhaus (1981, §3)	$n^2(\log q)\,(\log n)^{O(1)}$
Randomized solution; first $O(n)$-space polynomial-time complete factorization method.	
Schoof (1985)	$(\log p)^6(\log\log p)^{O(1)}$
$k = 1$ and $n = 2$: computes square roots mod p of constant integers, such as -1; running time stated as bit complexity. Only known deterministic factorization method that is *unconditionally* of polynomial running time as a function in $\log p$.	
Huang (1991a and 1991b)	$(n \log p)^{O(1)}$
$k = 1$, i.e., $\mathbb{F}_q = \mathbb{Z}/(p)$ and the given pre-image polynomial with integer coefficients has an Abelian Galois group over $\mathbb{Q}$. The running time is conditional on the validity of the ERH. See Rónyai (1989) for the generalization to the case where the Galois group can be arbitrary of cardinality $n^{O(1)}$.	
von zur Gathen and Shoup (1991)	$(n^2 + n\log q)\,(\log n)^{O(1)}$
Randomized solution with $O(n\sqrt{n})$ space requirement. If all irreducible factors have the same degree, the time drops to $(n^{(\omega+1)/2} + n\log q)(\log n)^{O(1)}$ and the space to $O(n)$.	

Shoup (1990) reduce the number of random bits needed, up to the point where trying all choices produces a $\sqrt{p}(n \log p)^{O(1)}$ algorithm (Shoup 1990b). The latter complexity has also been achieved by H. W. Lenstra, Jr., through an adaptation of the Pollard-Strassen integer factoring method.

If one represents the Galois field $\mathbb{F}_q$, $q = p^k$, as the polynomial algebra $\mathbb{F}_p[z]/(g(z))$, where g is an irreducible polynomial of degree k over the fixed field $\mathbb{F}_p$, it becomes intriguing to investigate the complexity of factoring over $\mathbb{F}_q$ in terms of the input degrees n and k. Efficient methods are described by Thiong ly (1989) and Shoup (1991b). The asymptotically fastest deterministic method is at this time due to von zur Gathen and Shoup (1991). A related problem is the *deterministic* construction of an irreducible polynomial of degree n over $\mathbb{F}_p$. The first solution that has running time a polynomial function in n and p seems to be Chistov's (1984) (see also Shoup 1990b and the survey by Shparlinskiy 1990).

3 Univariate Polynomials over Fields of Characteristic Zero

3.1 The Rational Numbers

The dramatic events leading to the polynomial-time factorization algorithm for $\mathbb{Q}[x]$ (Lenstra et al. 1982) are described in proper detail in (Kaltofen 1990a). It appears that I have coined the name "Berlekamp-Hensel algorithm" for the basic algorithm that factors a rational polynomial first modulo a prime p and then lifts to a factorization modulo a sufficiently high power p^k before recovering the rational factors (Kaltofen 1982). I now prefer the more appropriate name "*Berlekamp-Zassenhaus algorithm*," as this technique was introduced by Zassenhaus (1969). In this algorithm exponentially many steps may occur due the possibility of so-called "extraneous" factor images which do not correspond to rational factors. The French literature calls such factors "*parasitic*," a notion which I now also prefer. Indeed, parasitic factorizations are closely related to what mathematicians refer to as higher reciprocity laws and non-Hilbertian fields. The polynomial-time method by Lenstra et al. (1982) was called by Susan Landau the "L^3 (pronounced: L-cubed)" algorithm, a name which apparently stuck.

I sometimes read that when it comes to factoring an actual polynomial, the Berlekamp-Zassenhaus approach is always the successful choice. This is not so: Monagan (1986) has demonstrated that by finding integer values for the variable of certain irreducible polynomials with integer coefficients which produce, when evaluating the polynomials at those values, integers with sufficiently large prime factors, one may exhibit the irreducibility of the given polynomials; however, that would be quite hopeless by the Berlekamp-Zassenhaus method. Such a reduction to integer factoring was theoretically analyzed by Adleman and Odlyzko (1983).

The time for the L^3 algorithm out-performing the Berlekamp-Zassenhaus method in especially nasty cases — products of so-called Swinnerton-Dyer polynomials (Kaltofen et al. 1983) — also seems to have arrived. In our studies of defining equations of class fields (Kaltofen and Yui 1991) we have implemented a version of an algorithm that finds the minimum polynomial from a high precision approximation of one of its complex roots (Schönhage 1984, Kannan et al. 1988). Imin Chen has implemented in the programming language C a version of the lattice reduction algorithm that avoids rational number arithmetic by keeping track of the numerators $\kappa_{i,j} \in \mathbb{Z}$ of the $\mu_{i,j} = \kappa_{i,j}/d_j \in \mathbb{Q}$ (see Lenstra et al. 1982 for the definition of the quantities $\mu_{i,j}$ and d_j) and which is based on the long integer arithmetic of the PARI system (Batut et al. 1991). On a Sun 4 workstation the minimum polynomial of degree 47 of a real algebraic number has been computed in about 31 hours.

A problem somewhat related to polynomial factorization is *functional decomposition* of a polynomial: given $f(x) \in \mathsf{K}[x]$, where K is a field, find polynomials $g(x) = a_s x^s + \cdots + a_1 x + a_0, h(x) \in \mathsf{K}[x]$ such that

$$f(x) = g(h(x)) = a_s h(x)^s + \cdots + a_1 h(x) + a_0.$$

For $\mathsf{K} = \mathbb{Q}$, Kozen and Landau (1989) discovered a surprisingly simple solution of quadratic running time in the degree of f. Their solution was further improved and generalized by von zur Gathen (1990a and 1990b). I also refer to the latter papers for a comprehensive bibliography to previous work. Most recently, Zippel (1991) has investigated the problem of *rational function* decomposition.

3.2 Algebraic Number Fields

It is known at least since Kronecker (1882) that factoring polynomials over finite algebraic extensions of $\mathbb{Q}$ can be reduced to factoring polynomials over $\mathbb{Q}$. Kronecker introduces a transcendental element, hence increases the number of variables by one, which Trager (1976) shows how to avoid. The latter method is analyzed in Chistov and Grigoryev (1982) and in Landau (1985).

Here the question of representing algebraic elements arises. Abbott et al. (1986) demonstrate that it can be advantageous to keep multiple algebraic extensions $\mathbb{Q}(\alpha_1, \ldots, \alpha_s)$ rather than construct a simple extension $\mathbb{Q}(\vartheta)$. Arithmetic in such fields can be also speeded by modular imaging. In particular, we point to work by Weinberger and Rothschild (1976), by Lenstra (1983) and (Abbott 1988), and by Smedley (1989) (see also Geddes et al. (1988)). These methods usually lead to faster ways of factoring polynomials over algebraic extensions than the Kronecker reduction, since the latter produces inadvertently bad inputs for the Berlekamp-Zassenhaus method.

3.3 The Complex Numbers and P-adic Fields

The well-studied problem of finding complex roots of rational, say, polynomials, can also be considered a polynomial factorization task. Here we are not considering numerical methods but those that can for a given precision $2^{-l}, l > 0$, find complex rational numbers $a_j + \mathbf{i}\, b_j$, $a_j, b_j \in \mathbb{Q}$, such that for the roots $\zeta_j \in \mathbb{C}$ we have $|\zeta_j - (a_j + \mathbf{i}\, b_j)| < 2^{-l}$. One such method, invented in the last century by Routh and Hurwitz, is based on the Cauchy principal of argument and Sturm sequences (see Marden 1949, Pinkert 1976, and Wilf 1978). Collins (1977) observed that the computation of the necessary Sturm sequences can be replaced by any real root isolation method. A different modification computes the change of arguments around the nested contours by approximate complex integration (Schönhage 1982). A further issue is to find well-balanced splits of the root clusters, so that each sub-division contains sufficiently many complex roots. All these ideas can be combined to prove that arbitrary high precision approximations can be computed within the theoretically important parallel complexity class $\mathcal{NC}$ (Neff 1990; see also Ben-Or and Tiwari 1990 building on earlier work with Feig and Kozen; see Cook 1985 for a discussion of the complexity model $\mathcal{NC}$). We wish to point out that multidimensional Newton iteration (see, e.g., Linwood 1990 or Kerner 1966) seems computationally far more efficient than any of the "infallible" methods mentioned above. However, the question of how to compute a suitable starting point for the Newton iteration appears unsolved (see §7).

A similar problem is the computation of p-adic approximations to the p-adic factors of an integer, say, polynomial. By the Hensel lemma, the difficult case is when the prime radix divides the discriminant of the polynomial. We merely refer to the papers by Zassenhaus (1975), Trotter (1982), and Chistov (1987) for a discussion of this case.

4 Polynomials in Two Variables

Although the title of this section may suggest otherwise, all but two algorithms disussed in this section generalize to several variables, the exceptions being the algorithms by Duval (1991) and Bajaj et al. (1989) on algebraic curves. However, the running times of the other algorithms, when applied to an input with n variables, are exponential in n, which is only acceptable if the input or output also has a comparable size growth. As explained in §1, such a dense representation model becomes exceedingly unrealistic as n increases. In that situation, the approach discussed in §5 ought to be taken.

4.1 Finite Extensions of Prime Fields

At least three decidedly distinct methods are known for factoring densely represented multivariate polynomials over $\mathbb{Q}$ or finite algebraic extensions in polynomial-time. One is a generalization of the lattice reduction method (Lenstra 1987). Chistov's and Grigoryev's (1982) algorithm combines a modular version of an effective Hilbert irreducibility theorem (Kaltofen 1985a, §7) with a polynomial-time lattice-reduction based algorithm for factoring in $\mathbb{F}_q[x, y]$, and thus determines the proper combinations of parasitic univariate rational factor images quickly. A third approach (Kaltofen 1985a) applies Zassenhaus's (1981) root approximation scheme, which reduces the problem to univariate factorization. That algorithm has the advantage that it can be formulated for an abstract coefficient field, provided a univariate polynomial factorization method over that field is also given. This universality has yielded a plethora of other results (see §4.2).

The problem of factoring in $\mathbb{F}_q[x, y]$ is polynomial-time equivalent as a function in the input degree and $\log q$ to the problem of factoring in $\mathbb{F}_q[x]$ (Chistov and Grigoryev 1982, §3). Hence for large characteristic any known polynomial-time solution requires randomization; but in testing polynomials in $\mathbb{F}_q[x, y]$ for irreducibility randomization can be avoided (Kaltofen 1987).

It is not completely clear to me on which inputs any of these methods yields a procedure on a computer that is practically superior to the multivariate Berlekamp-Zassenhaus method (Musser 1975, Yun 1974, Wang 1978, and von zur Gathen 1984). The reason is that by virtue of the orginal Hilbert irreducibility theorem, parasitic factors are rare, although no complete mathematical justification for this phenomenon seems to be known (c.f. Sprindžuk 1983). An even more special case is the problem of factoring in $\mathbb{F}_q[x, y]$. The univariate Berlekamp-Zassenhaus approach can be taken with lifting in the domain $\mathbb{F}_q[[y]]$, and combining the appropriate parasitic factors can then be accomplished quite reasonably (see Viry 1990).

Nevertheless, the polynomial-time algorithms are not a purely theoretical feat. If one needs to factor a multivariate polynomial over the complex numbers, say, the univariate image in the Berlekamp-Zassenhaus method is guaranteed to factor into linear polynomials, that is, one *always* produces the maximum possible number of parasitic factors and the algorithm requires, on absolutely irreducible inputs, surely at least exponential time. How one can avoid such computational explosion is discussed next.

4.2 Algebraically Closed Fields

It is possible to concisely characterize an algebraic extension of the input polynomial's field of coefficients that contains all coefficients of any *individual* absolutely irreducible factor. Note that the minimum common super-field for all factors can be an extension of degree $d\,!$, where d is the degree of the input, as this is already the case for univariate polynomials. Several persons, among them Chistov and Grigoryev (1983, Lemma 1), Trager (1984, §3.2), Kaltofen (1985b, Theorem 1), and Dvornicich and Traverso (1987) have discovered the following simple but powerful lemma.

Lemma. *Let $f(x, y) \in \mathsf{K}[x, y]$, where K is a field of characteristic zero, be irreducible over K and monic in x, that is, with a lead monomial $x^{\deg(f)}$. Let $g(x, y) \in \overline{\mathsf{K}}[x, y]$, where $\overline{\mathsf{K}}$ is the algebraic closure of K, be an absolutely irreducible monic factor of f. Then there exists a root $\zeta \in \overline{\mathsf{K}}$ of $f(x, 0)$ such that the field generated by the coefficients of g is isomorphic to a sub-field of $\mathsf{K}(\zeta)$.*

Monicity of f can be achieved by a generic transformation of coordinates $y = ax + y'$, $a \in \mathsf{K}$. Note that irreducibility of f implies that the degree of g must divide the degree of f, which can be exploited (Yokoyama et al. 1990). Not only can one prove this lemma from the root approximation algorithm, but that algorithm also yields several additional properties (Kaltofen 1991). For one, the irreducibility condition for f may be replaced with the condition that the Sylvester resultant

$$\mathrm{Resultant}_x(f(x, 0), \partial f(x, 0)/\partial x) \neq 0,$$

which is true in characteristic zero if f has no square factor. Then, once the field $\mathsf{L} = \mathsf{K}(\zeta)$ is constructed, all remaining work can be carried out by arithmetic operations in $\mathsf{K}(\zeta)$. It has to be recognized that, generally, different roots $f(\zeta_1, 0) = f(\zeta_2, 0) = 0$ may produce the same absolutely irreducible factor. Of course, the question on how to identify such collisions and, for example, correctly count the number of factors is intimately connected with the representation of the fields $\mathsf{L}_1 = \mathsf{K}(\zeta_1)$ and $\mathsf{L}_2 = \mathsf{K}(\zeta_2)$ themselves.

One possible solution is to factor $f(z, 0) = \phi_1(z) \cdots \phi_r(z)$ such that ϕ_i are irreducible factors in $\mathsf{K}[z]$. Then we can choose the Kronecker model (see, for instance, Loos 1982)

$$\mathsf{L}_i = \mathsf{K}[z]/(\phi_i(z)), \quad \zeta_i = z \bmod \phi_i(z), \tag{1}$$

and represent elements in L_i in the vector algebra spanned by $\{1, z, z^2, \ldots, z^{\deg(\phi_i)-1}\}$ over K. Notice that all conjugates of ζ_i are represented in this way. With this representation the arithmetic in L_i can be reduced to arithmetic in K. By use of a polynomial factoring algorithm in $\mathsf{K}[z]$ and arithmetic in K we now may diagnose that the same absolutely irreducible factor is produced by two different ϕ_1 and ϕ_2, or by different roots of one ϕ_i (Kaltofen 1990b, Remarks in §2 after the algorithm Factorization over the Algebraic Closure).

For the concrete ground field $\mathsf{Q} = \mathsf{K}$, other representations of ζ are useful. For instance, in addition to the factor ϕ_i, one may associate with ζ a high precision complex rational approximation to a root of ϕ_i. The absolutely irreducible factors can then be converted to complex rational polynomials that approximate the actual complex factors. By appealing to factor coefficient separation results and by choosing very precise approximations, not only double factors but also entirely real factors or complex conjugate ones can be discerned (Kaltofen 1990b).

Emmy Noether (1922) established that the problem of deciding whether a polynomial is already absolutely irreducible requires no arithmetic in an algebraic extension and can be decided by arithmetic in K itself. The reader is also referred to Schmidt (1976) for an account of her approach written in English. This approach establishes that the coefficients of those polynomials of a given degree that factor over the algebraic closure form an algebraic variety of an ideal generated by polynomials with *integer* coefficients, the irreducibility forms. In (Kaltofen 1991) we give the following effective construction.

Theorem. *Let $d \geq 2$ and $n \geq 2$; there exist $2^{(d+n)^{O(1)}}$ polynomials*

$$\Phi_t(\ldots, C_{e_1,\ldots,e_n}, \ldots) \in \mathbb{Z}[\ldots, C_{e_1,\ldots,e_n}, \ldots],$$

$e_i \geq 0$, $e_1 + \cdots + e_n \leq d$, such that for any field K and any polynomial

$$f(X_1, \ldots, X_n) = \sum_{0 \leq e_1 + \cdots + e_n \leq d} c_{e_1,\ldots,e_n} X_1^{e_1} \cdots X_n^{e_n}$$

in $\mathsf{K}[X_1, \ldots, X_n]$ we have

$$\forall\, t\colon \Phi_t(\ldots, c_{e_1,\ldots,e_n}, \ldots) = 0 \iff f \text{ is reducible over } \overline{\mathsf{K}}, \text{ or } \deg(f) < d.$$

If K has positive characteristic p, the coefficients of Φ_t are to be taken modulo p in the left-hand-side equality. Furthermore, for all t,

$$\deg(\Phi_t) \leq 12d^6 \quad \text{and} \quad \|\Phi_t\|_1 \leq (2d)^{12d^7 + 12d^6 n + 32d^6},$$

where $\|\Phi_t\|_1$ denotes the sum of absolute values of the integral coefficients of Φ_t.

This fundamental theorem has many consequences in the theory of factorization over the algebraic closure. There are no reducible polynomials arbitrarily close to an absolutely irreducible one, and effective separation is possible (Kaltofen 1991, Theorem 10). Modular projection of the coefficients preserves absolute irreducibility for all but a finite number of exceptions, and these exceptions can be effectively bounded (Kaltofen 1991, Theorems 8 and 9).

The Kronecker representation (1) requires the factorization of $f(z, 0)$ and therefore does not yield an absolute irreducibility test based on arithmetic alone. Already in (Kaltofen 1985) we have removed this requirement at least if the field characteristic is zero. Essentially, we perform the algorithm simultaneously for all L_i, that is, in

$$\mathsf{K}[z]/(f(z,0)) \cong \mathsf{K}[z]/(\phi_1(z)) \otimes \cdots \otimes \mathsf{K}[z]/(\phi_r(z))$$

in place of in each separate L_i. This model of algebraic number field arithmetic has been formalized by Dicrescenzo and Duval (1987): an algebraic number ζ is represented by a not necessarily irreducible, but square-free, defining equation

$$\psi(z) \in \mathsf{K}[z], \quad \psi(\zeta) = 0. \tag{2}$$

Elements $\beta \in \mathsf{K}(\zeta)$ are represented as elements in the algebra $\mathsf{K}[z]/(\psi(z))$. The element β is not zero if $\mathrm{GCD}_z(\beta, \psi) = 1$, interpreting β as an element in $\mathsf{K}[z]$. In that case, β can be inverted by computing the Euclidean scheme $\sigma\beta + \tau\psi = 1$, $\sigma, \tau \in \mathsf{K}[z]$, yielding $\beta^{-1} = \sigma$. If β and ψ are not relatively prime in $\mathsf{K}[z]$, then, according to Dicrescenzo and Duval (loc. cit.), the computation has to split. If ζ is a root of $\mathrm{GCD}(\beta, \psi)$, then the element β is zero, otherwise it is not zero. In both cases, we obtain new defining equations for ζ. We call this representation of algebraic number fields the *lazy factorization model*. Indeed, Duval (1991) gives a procedure based on the geometry of the plane curve determined by $f(x, y) = 0$ that can count the number of irreducible components by use of the lazy factorization model.

I derive the above theorem by analyzing that variant of my algorithm for absolute irreducibility testing which executes that algorithm on generic inputs, that is polynomials whose coefficients are indeterminates. Polynomial degree and coefficient size bounds for all intermediately computed generic rational function field elements are obtained (Kaltofen 1991). These estimates then make it possible to formulate the *bit complexity* of my algorithms without specifying the actual coefficient field. For if we know that all arithmetic operations, that is $+$, $-$, $\times$, $\div$, and $= 0?$, on field elements of polynomial size is of polynomial-time bit complexity, then we can conclude that factoring algorithm using the lazy factorization model, for example, is also of polynomial-time bit complexity, that is the size of all intermediate values stay polynomially bounded. Of course, the actual element representation must be *canonical* with respect to the arithmetic operations. A famous counter-example otherwise is that of computing polynomial GCDs using quotient field arithmetic without reducing numerators and denominators by a common factor, which results in exponential bit complexity for an algorithm with quadratic arithmetic complexity (Knuth 1981, §4.6.1, Eq. (27)).

Algebraic elements in the lazy factorization representation (2) are difficult to compare if they are computed by different branches. To illustrate this fact, consider $\psi(z) = (z^2 - 2)(z^2 - 18)$: assume the computation has split and the resulting algebraic numbers of the two branches are $z_1 \bmod z_1^2 - 2$ and $z_2/3 \bmod z_2^2 - 18$. There are conjugates of both defining equations such that the resulting algebraic numbers both represent $\sqrt{2}$. By associating with ζ an element $\widetilde{\zeta}$ that allows the distinction of conjugates of ψ this problem may be remedied. Note that the element $\widetilde{\zeta}$ depends on the field K: if K is an algebraic number field, we can use a rational complex number that isolates a root of ψ; if K is a function field, we may use a truncated Puiseux series.

An element $\beta \in \mathsf{K}(\zeta)$ is now represented by the triple

$$\chi(z) \in \mathsf{K}[z], \quad \psi(z) \in \mathsf{K}[z], \quad \widetilde{\zeta} \tag{3}$$

with $\beta = \chi(\zeta)$. As in the lazy factorization model, zero-testing requires the GCD computation $\gamma(z) = \mathrm{GCD}(\chi(z), \psi(z))$. Then $\beta = 0$ if and only if $\gamma(\zeta) = 0$, which can be checked using the approximate root $\widetilde{\zeta}$. Non-zero elements can be inverted again using the Euclidean scheme $\sigma\chi + \tau\psi = \gamma$, and we get as the representation of β^{-1} the triple $\sigma(z)$, $\psi(z)/\gamma(z)$, and $\widetilde{\zeta}$. Arithmetic now has changed from the lazy factorization model in that the operands may have different defining equations, say $\beta_1 = \chi_1(\zeta)$ with $\psi_1(\zeta) = 0$ and $\beta_2 = \chi_2(\zeta)$ with $\psi_2(\zeta) = 0$. We can compute $\psi_3(z) = \mathrm{GCD}(\psi_1(z), \psi_2(z))$ and then perform the arithmetic operations modulo ψ_3. We call the representation (3) the *single path lazy factorization* model for the field $\mathsf{K}(\zeta)$ (Kaltofen 1991 and Lombardi 1989).

A salient feature of the single path lazy factorization model is that the factorization problem over the algebraic closure can then be solved in parallel computational models, such as the theoretically important complexity class $\mathcal{NC}$. For $\mathsf{K} = \mathbb{Q}$, poly-log bit complexity is accounted for in part by my generic analysis discussed before, and in part by Neff's (1990) parallel complex root approximation algorithm mentioned in §3.3. Furthermore, this theory immediately generalizes to function fields $\mathbb{Q}(u)$. Entirely sequential factoring methods over function fields are also discussed in Chistov (1987). Precursory $\mathcal{NC}$-results are found in Kaltofen (1985b) and Bajaj et al. (1989). The latter paper investigates the geometry of the four dimensional real two-fold corresponding to the curve $f(x, y) = 0$.

5 Polynomials in Many Variables

When the problem size may depend on the number of variables, a key issue is the representation of the input polynomials and their irreducible factors. For example, a classical such problem, Frobenius's (1896) original approach to group characters, is the factorization of the determinant of the multiplication table of a finite group whose elements are denoted by n variables. Clearly, such a determinant has potentially $\binom{2n-1}{n} > 2^{2n-2}/n$ many monomials, but it may be represented by a division-free straight-line program of length $O(n^4 \log n)$ which computes its value for any value of its variables. The theory of efficient manipulation of polynomials in straight-line representation (Kaltofen 1988 and 1989) proves that then all suitably normalized irreducible factors can also be represented by polynomial-sized straight-line programs. Furthermore, these straight-line programs can be constructed in random polynomial-time.

All algorithms known with this flavor utilize so-called effective Hilbert irreducibility theorems that probabilistically prevent the occurrence of parasitic factors (Heintz and Sieveking 1981, von zur Gathen 1985, Kaltofen 1985a, Bajaj et al. 1989, Kaltofen 1991). In the last reference the following estimate is proven.

Theorem. *Let K be a perfect field, $g \in \mathsf{K}[X_1, \ldots, X_n]$. If the elements $a_1, \ldots, a_n$, $b_1, \ldots, b_n$, $c_2, \ldots, c_n \in S \subset \mathsf{K}$ are randomly and uniformly selected from the set S, then the probability*

$$\begin{aligned}&\mathrm{Prob}(g(x + a_1, b_2x + c_2y + a_2, \ldots, b_nx + c_ny + a_n)\\&\qquad \textit{has the same number of irreducible factors over } \mathsf{K} \textit{ as } g) \geq 1 - 2\deg(g)^4/\mathrm{card}(S),\end{aligned}$$

where $\mathrm{card}(S)$ denotes the cardinality of S.

We have implemented the algorithm for factoring polynomials given by straight-line programs (Freeman et al. 1988). Although we then could factor polynomials inaccessible to any other factorization method, it was observed that the straight-line answers become larger and larger. The explanation is simply the fact that the size of the answer is related to the time it takes to compute it, since a part of the algorithm performs Hensel lifting by encoding it in the answer. For instance, the quadratic factor of a 16×16 group determinant is found as a straight-line program with 199,732 instructions.

Fortunately, our work on the so-called *black-box* representation (Kaltofen and Trager 1990) provides a way out of this predicament: a black box is an object which takes as input a value for each variable, and then produces the value of the polynomial it represents at the specified point (see Figure 1).

The algorithm's outputs are procedures which will evaluate all irreducible factors at arbitrary points (supplied as the input). These procedures make oracle calls to the black

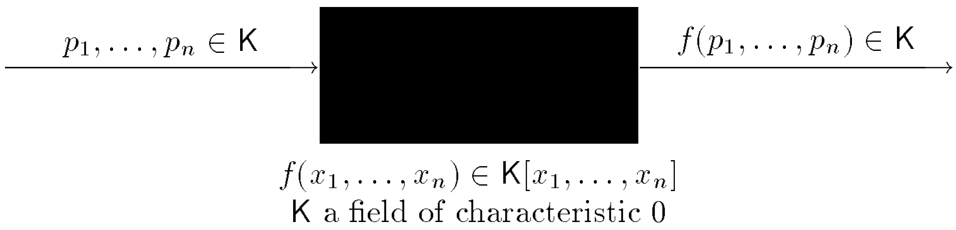

Figure 1: Black box representation of a polynomial.

box given as the input to the algorithm to evaluate them at certain points dependent on the inputs to the procedure (see Figure 2). It is, of course, crucial that the program for the irreducible factors evaluates a fixed associate for each multivariate factor. Moreover, the program is with controllably high probability correct, that is it then will always return the correct evaluations of the factors, independently of the input values an adversary may have chosen for the variables.

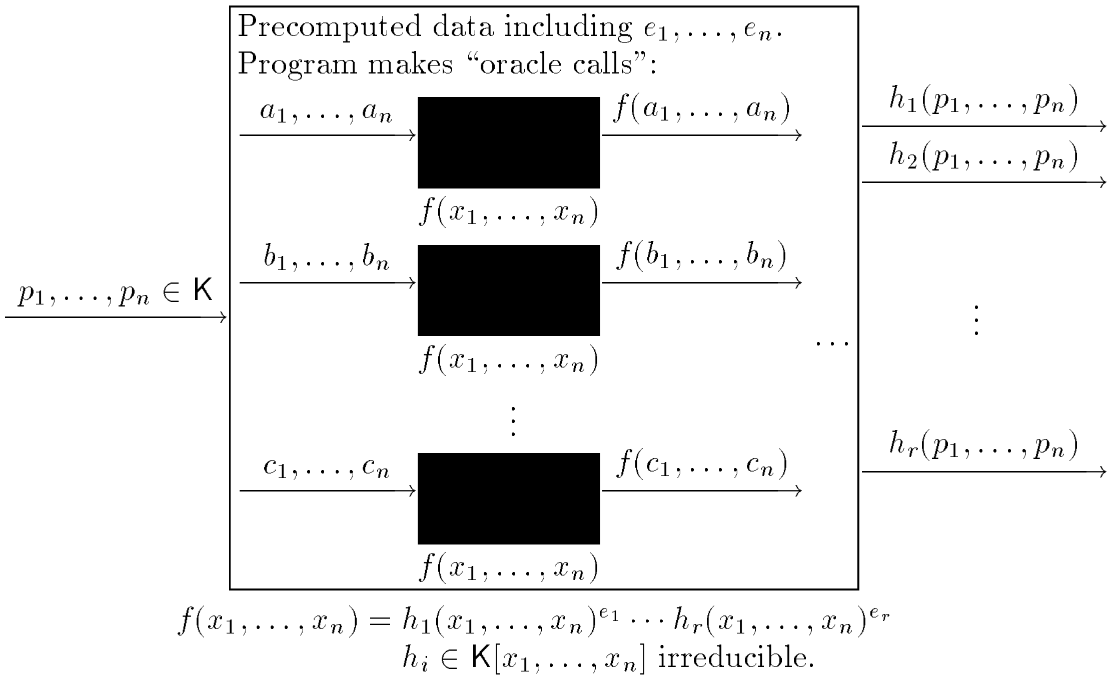

Figure 2: The program for evaluating the irreducible factors of a black box polynomial.

Our constructed programs with oracle calls to the input black box are much more space efficient than the straight-line program anwers. Both can be rapidly converted to sparse format using any of the new sparse polynomial interpolation algorithms (Ben-Or and Tiwari 1988, Grigoryev et al. 1990, Zippel 1979 and 1990, Kaltofen et al. 1988 and 1990). For polynomials given by black boxes, the sparse interpolations of the factors can be easily distribute over a series of processors, since the amount of data that needs to be sent to the individual processors is very small. Precisely this strategy has motivated our design of the DSC distributed symbolic computing tool (Diaz et al. 1991).

6 Implementations and Applications

As said in §1, any of the general purpose symbolic computation systems provide a polynomial factorization facility, mostly based on the Berlekamp-Zassenhaus approach. On the design of the factorizer in Macsyma the reader is referred to Wang (1978), on the one in Reduce to Norman and Moore (1981), and the one in Axiom a.k.a. Scratchpad to Lucks (1986) (see also Davenport et al. 1991). The latter was used to factor a univariate polynomial of degree 388 with rational coefficients with numerators and denominators up to 200 decimal digit long, which arose as an intermediate result in one of Gebauer's Gröbner basis reductions. The factorization into 4, 16, 32, 56, 64, 80, and 128 degree factors could be carried out on an IBM 3081D in a few hours.

One of the first study of sparse factorizations is by Claybrook (1976). Zippel (1981) has also implemented his method. The implementation of a factorizer based on straight-line program representation is described in Freeman et al. (1988), with which one of the largest polynomials ever was factored: a 16×16 group determinant, which if expanded would have over 300 million terms. The factorization and subsequent conversion of the single quadratic factor to sparse format took several hours on a Symbolics 3600 Lisp machine. One of the few careful implementations of multivariate factorization over algebraic function fields is described in Abbott (1988). This algorithm forms a building block of closed-form integration methods of algebraic functions. The problem of factoring in $\mathbb{F}_q[x]$ is somewhat more specialized. Different implementations are discussed in Menezes et al. (1988), Wang (1990), and Trevison and Wang (1991); the first reference comes from Berlekamp's original motivation: coding theory.

Large scale polynomial factorizations arise in several problem solving methodologies. The greatest impetus is perhaps given by the solution of non-linear algebraic equations by means of Gröbner basis reduction: in certain situations, only the factorization of an intermediate polynomial can make such an approach feasible (Melenk et al. 1988). Furthermore, factorizations over the reals and complex numbers are useful in geometric modeling. E.g., the algorithm for parameterizing an implicitly given curve by Abhyankar and Bajaj (1988) requires an absolutely irreducible input. The applications to Galois theory, e.g., by Landau and Miller (1985) and by Landau (1989), are quite classical.

7 Open Problems

Before I discuss several challenging unsolved problems, which illustrate the vitality of research on the subject of polynomial factorization, I shall give a brief update on the open problems raised in Kaltofen (1990a, §6). On the questions of, within the complexity class $\mathcal{NC}$, factoring polynomials in standard representation in $\mathbb{Q}[x]$, or testing for irreducibility in $\mathbb{F}_p[x]$, where p is large, and in $\mathbb{Q}[x, y]$, no progress can be reported. However, several new $\mathcal{NC}$-complexity results have been obtained in the meantime; see §3.3 and §4.2. Also, the following problem remains open: given is a straight-line program of length l that computes a multivariate polynomial. Are then all irreducible factors of degree polynomial in l computable by straight-line programs of length also polynomial in l? Nonetheless, Lickteig (1990) could establish polynomial straight-line complexity for such factors in the weaker approximative algebraic complexity model.

Problem 1 (von zur Gathen and Shoup 1991): Given a polynomial of degree n in $\mathbb{F}_p[x]$, can one find with randomization all factors in $(n^\kappa + n \log p)(\log n)^{O(1)}$, $\kappa < 2$, expected arithmetic operations in $\mathbb{F}_p$?

Problem 2 (communicated by M. Karpinski and L. Lovász): Given integers $C_0, \ldots, C_t$ and exponents l, $E_1, \ldots, E_t$ with $|C_i| < 2^l$, $E_j < 2^l$, can one determine in time polynomial in l whether the polynomial

$$C_0 + C_1 x^{E_1} + C_2 x^{E_2} + \cdots + C_t x^{E_t}$$

has a real root?

Problem 3 (communicated by B. Sturmfels): From the support vectors $(e_{j,1}, \ldots, e_{j,n})$ of a sparse multivariate polynomial

$$\sum_{j=1}^{t} c_{e_{j,1},\ldots,e_{j,n}} X_1^{e_{j,1}} \cdots X_n^{e_{j,n}},$$

compute by geometric considerations the support vectors of all possible factorizations. Note that Dumas's irreducibility criterion (van der Waerden 1953, §24) is an example of such arguments.

Problem 4: Consider the symmetric functions

$$x^n + \sum_{i=1}^{n} (-1)^i \sigma_i(z_1, \ldots, z_n) x^{n-i} = \prod_{j=1}^{n} (x - z_j).$$

For n complex rational coefficients $c_1, \ldots, c_n$, consider the system of algebraic equations

$$c_i = (-1)^i \sigma_i(z_1, \ldots, z_n), \quad 1 \le i \le n. \tag{4}$$

Give a fast, infallible method to compute initial complex rational values ζ_j such that the multidimensional Newton iteration started at the initial points $z_j^{(0)} = \zeta_j$ converges for the system (4) to the roots of $x^n + c_1 x^{n-1} + \cdots + c_n$ (c.f. Linwood 1990 and Pasquini and Trigiante 1985).

Problem 5: Given $f(x, y) \in \mathbb{Q}(\mathbf{i})[x, y]$ monic in x and absolutely irreducible, can one then decide in polynomial time in the degree and coefficient size of f, and the precision $l > 0$, whether there exists a factorizable polynomial $\widetilde{f}(x, y) \in \mathbb{C}[x, y]$ all of whose coefficients are in absolute distance within 2^{-l} of the corresponding coefficients of f? in mathematical terms: $\exists \widetilde{f} \in \mathbb{C}[x, y]$ absolutely reducible such that $\|f - \widetilde{f}\|_\infty \le 2^{-l}$. A solution of this problem may lead to a method of factoring polynomials whose coefficients are given imprecisely by complex floating point numbers.

Acknowledgement: I wish to express my appreciation to Joachim von zur Gathen, Hendrik Lenstra, Jr., Victor Shoup, and Igor Shparlinskiy for their input.

Literature Cited

Abbott, J. A., "Factorization of polynomials over algebraic function fields," *Doctoral Thesis*, Univ. Bath, England, 1988.

Abbott, J. A., "Recovery of algebraic numbers from their p-adic approximations," *Proc. ACM-SIGSAM 1989 Internat. Symp. Symbolic Algebraic Comput.*, pp. 112–120 (1989).

Abbott, J. A., Bradford, R. J., and Davenport, J. H., "The Bath algebraic number package," *Proc. 1986 ACM Symp. Symbolic Algebraic Comp.*, pp. 250–253 (1986).

Abhyankar, S. and Bajaj, C., "Automatic rational parameterization of curves and surfaces III: Algebraic plane curves," *Computer Aided Geometric Design* **5**, pp. 308–321 (1988).

Adleman, L. M., Manders, K., and Miller, G. L., "On taking roots in finite fields," *Proc. 18th IEEE Symp. Foundations Comp. Sci.*, pp. 175–178 (1977).

Adleman, L. M. and Odlyzko, A. M., "Irreducibility testing and factorization of polynomials," *Math. Comp.* **41**, pp. 699–709 (1983).

Ankeny, N. C., "The least quadratic non residue," *Ann. of Math.* **55**/1, pp. 65–72 (1952).

Arwin, A., "Über die Kongruenzen von dem fünften und höheren Graden nach einem Primzahlmodulus," *Arkiv f. matematik, astronom. o. fysik* **14**, pp. 1-46 (1918). In German.

Bach, E. and Shoup, V., "Factoring polynomials using fewer random bits," *J. Symbolic Comput.* **9**/3, pp. 229–239 (1990).

Bajaj, C., Canny, J., Garrity, T., and Warren, J., "Factoring rational polynomials over the complexes," *Proc. ACM-SIGSAM 1989 Internat. Symp. Symbolic Algebraic Comput.*, pp. 81–90 (1989).

Batut, C., Bernardi, D., Cohen, H., and Olivier, M., "User's Guide to PARI-GP," *Manual*, February 1991.

Ben-Or, M., "Probabilistic algorithms in finite fields," *Proc. 22nd IEEE Symp. Foundations Comp. Sci.*, pp. 394–398 (1981).

Ben-Or, M. and Tiwari, P., "A deterministic algorithm for sparse multivariate polynomial interpolation," *Proc. 20th Annual ACM Symp. Theory Comp.*, pp. 301–309 (1988).

Ben-Or, M. and Tiwari, P., "Simple algorithms for approximating all roots of a polynomial with real roots," *J. Complexity* **6**, pp. 417–442 (1990).

Berkowitz, S. J., "On computing the determinant in small parallel time using a small number of processors," *Inform. Process. Letters* **18**, pp. 147–150 (1984).

Berlekamp, E. R., "Factoring polynomials over finite fields," *Bell Systems Tech. J.* **46**, pp. 1853–1859 (1967). Republished in revised form in: E. R. Berlekamp, *Algebraic Coding Theory*, Chapter 6, McGraw-Hill Publ., New York 1968.

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

Butler, M. C. R., "On the reducibility of polynomials over a finite field," *Quart. J. Math., Oxford Ser. (2)* **5**, pp. 102–107 (1954).

Camion, P., "Un algorithme de construction des idempotents primitifs d'ideaux d'algebres sur $\mathbb{F}_q$," *Ann. Discrete Math* **12**, pp. 55-63 (1982).

Cantor, D. G. and Zassenhaus, H., "A new algorithm for factoring polynomials over finite fields," *Math. Comp.* **36**, pp. 587–592 (1981).

Chistov, A. L., "The construction of a finite field in polynomial time," *Proc. 7 All-Union Conf. on Math. Logic (Novosibirsk)*, p. 196 (1984). In Russian.

Chistov, A. L., "Efficient factorization of polynomials over a local field," *Soviet Math. Doklady (AMS Translation)* **37**/2, pp. 430-433 (1987).

Chistov, A. L. and Grigoryev, D. Yu., "Polynomial-time factoring of multivariable polynomials over a global field," *LOMI Preprints* **E-5-82**, USSR Acad. Sci., Steklov Math. Inst., Leningrad, 1982.

Chistov, A. L. and Grigoryev, D. Yu., "Subexponential-time solving of systems of algebraic equations I," *LOMI Preprints* **E-9-83**, USSR Acad. Sci., Steklov Math. Inst., Leningrad, 1983.

Claybrook, B. G., "A new approach to the symbolic factorization of multivariate polynomials," *Artificial Intelligence* **7**, pp. 203–241 (1976).

Collins, G. E., "Quantifier elimination for real closed fields by cylindrical algebraic decomposition," in *Proc. 2nd GI Conf. Automata Theory Formal Lang.*, Springer Lec. Notes Comp. Sci. **33**; pp. 515–532, 1975.

Collins, G. E., "Infallible calculation of polynomial zeros to specified precision," in *Mathematical Software III*, edited by J. R. Rice; Academic Press, New York, pp. 35–68, 1977.

Cook, S. A., "A taxonomy of problems with fast parallel algorithms," *Inf. Control* **64**, pp. 2–22 (1985).

Davenport, J. H., Gianni, P., and Trager, B. M., "Scratchpad's view of algebra II: a categorical view of factorization," in *Proc. 1991 Internat. Symp. Symbolic Algebraic Comput.*, edited by S. M. Watt; ACM Press, pp. 32–38, 1991.

Diaz, A., Kaltofen, E., Schmitz, K., and Valente, T., "DSC A System for Distributed Symbolic Computation," in *Proc. 1991 Internat. Symp. Symbolic Algebraic Comput.*, edited by S. M. Watt; ACM Press, pp. 323-332, 1991.

Dicrescenzo, C. and Duval, D., "Le système D5 de calcul formel avec des nombres algébriques," *Chapter 1 of the Doctoral Thesis by D. Duval*, Univ. Grenoble, 1987. In French.

Duval, D., "Absolute factorization of polynomials: a geometric approach," *SIAM J. Comput.* **20**/1, pp. 1–21 (1991).

Dvornicich, R. and Traverso, C., "Newton symmetric functions and the arithmetic of algebraically closed fields," in *Proc. AAECC-5*, Springer Lect. Notes Comput. Sci. **356**; pp. 216–224, 1987.

Freeman, T. S., Imirzian, G., Kaltofen, E., and Lakshman Yagati, "DAGWOOD: A system for manipulating polynomials given by straight-line programs," *ACM Trans. Math. Software* **14**/3, pp. 218–240 (1988).

Fröhlich, A. and Shepherdson, J. C., "Effective procedures in field theory," *Phil. Trans. Roy. Soc., Ser. A* **248**, pp. 407–432 (1955/56).

Gantmacher, F. R., *The Theory of Matrices, Vol. 1*; Chelsea Publ. Co., New York, N. Y., 1960.

von zur Gathen, J., "Hensel and Newton methods in valuation rings," *Math. Comp.* **42**, pp. 637–661 (1984).

von zur Gathen, J., "Irreducibility of multivariate polynomials," *J. Comp. System Sci.* **31**, pp. 225–264 (1985).

von zur Gathen, J., "Factoring polynomials and primitive elements for special primes," *Theoretical Comput. Sci.* **52**, pp. 77–89 (1987).

von zur Gathen, J., "Functional decomposition of polynomials: the tame case," *J. Symbolic Comput.* **8**/3, pp. 281–299 (1990a).

von zur Gathen, J., "Functional decomposition of polynomials: the wild case," *J. Symbolic Comput.* **10**/5, pp. 437–452 (1990b).

von zur Gathen, J. and Shoup, V., "Computing Frobenius maps and factoring polynomials," in *Proc. 24th Ann. ACM Symp. Theory Comput.*; ACM Press, pp. 97–105, 1992.

Geddes, K. O., Gonnet, G. H., and Smedley, T. J., "Heuristic methods for operations with algebraic numbers," in *Proc. ISSAC '88*, Springer Lec. Notes Comput. Sci. **358**, edited by P. Gianni; pp. 475–480, 1988.

Grigoryev, D. Yu., Karpinski, M., and Singer, M. F., "Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields," *SIAM J. Comput.* **19**/6, pp. 1059–1063 (1990).

Heintz, J. and Sieveking, M., "Absolute primality of polynomials is decidable in random polynomial-time in the number of variables," *Proc. ICALP '81, Springer Lec. Notes Comp. Sci.* **115**, pp. 16–28 (1981).

Huang, M.-D. A., "Generalized Riemann hypothesis and factoring polynomials over finite fields," *J. Algorithms* **12**/3, pp. 464–481 (1991a).

Huang, M.-D. A., "Factorization of polynomials over finite fields and decomposition of primes in algebraic number fields," *J. Algorithms* **12**/3, pp. 482–489 (1991b).

Kaltofen, E., "Polynomial factorization," in *Computer Algebra, 2nd ed.*, edited by B. Buchberger et al.; Springer Verlag, Vienna, pp. 95–113, 1982.

Kaltofen, E., "Effective Hilbert irreducibility," *Information and Control* **66**, pp. 123–137 (1985c).

Kaltofen, E., "Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization," *SIAM J. Comp.* **14**/2, pp. 469–489 (1985a).

Kaltofen, E., "Fast parallel absolute irreducibility testing," *J. Symbolic Comput.* **1**, pp. 57–67 (1985b). Misprint corrections: *J. Symbolic Comput.* **9**, p. 320 (1989).

Kaltofen, E., "Deterministic irreducibility testing of polynomials over large finite fields," *J. Symbolic Comp.* **4**, pp. 77–82 (1987).

Kaltofen, E., "Greatest common divisors of polynomials given by straight-line programs," *J. ACM* **35**/1, pp. 231–264 (1988).

Kaltofen, E., "Factorization of polynomials given by straight-line programs," in *Randomness and Computation*, Advances in Computing Research **5**, edited by S. Micali; JAI Press, Greenwhich, Connecticut, pp. 375–412, 1989.

Kaltofen, E., "Polynomial Factorization 1982-1986," in *Computers in Mathematics*, Lecture Notes in Pure and Applied Mathematics **125**, edited by D. V. Chudnovsky and R. D. Jenks; Marcel Dekker, Inc., New York, N. Y., pp. 285–309, 1990a.

Kaltofen, E., "Computing the irreducible real factors and components of an algebraic curve," *Applic. Algebra Engin. Commun. Comput.* **1**/2, pp. 135–148 (1990b).

Kaltofen, E., "Effective Noether irreducibility forms and applications," *Tech. Rep.* **91-2**, Dept. Comput. Sci., Rensselaer Polytech. Inst., Troy, N. Y., January 1991. Extended abstract in *Proc. 23rd Ann. ACM Symp. Theory Comput.*, ACM Press, pp. 54–63 (1991).

Kaltofen, E. and Lakshman Yagati, "Improved sparse multivariate polynomial interpolation algorithms," *Proc. ISSAC '88, Springer Lect. Notes Comput. Sci.* **358**, pp. 467–474 (1988).

Kaltofen, E., Lakshman Y. N., and Wiley, J. M., "Modular rational sparse multivariate polynomial interpolation," in *Proc. 1990 Internat. Symp. Symbolic Algebraic Comput.*, edited by S. Watanabe and M. Nagata; ACM Press, pp. 135–139, 1990.

Kaltofen, E., Musser, D. R., and Saunders, B. D., "A generalized class of polynomials that are hard to factor," *SIAM J. Comp.* **12**/3, pp. 473–485 (1983).

Kaltofen, E. and Trager, B., "Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators," *J. Symbolic Comput.* **9**/3, pp. 301–320 (1990).

Kaltofen, E. and Yui, N., "Explicit construction of Hilbert class fields of imaginary quadratic fields by integer lattice reduction," in *Number Theory New York Seminar 1989–1990*, edited by D. V. Chudnovsky, G. V. Chudnovsky, H. Cohn, and M. B. Nathanson; Springer Verlag, New York, pp. 150–202, 1991.

Kannan, R., Lenstra, A. K., and Lovász, L., "Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers," *Math. Comp.* **50**, pp. 235–250 (1988).

Kerner, I. O., "Ein Gesamtschrittverfahren zur Berechnung der Nullstellen von Polynomen," *Numer. Math.* **8**, pp. 290–294 (1966). In German.

Knuth, D. E., *The Art of Computer Programming, Vol. 2, Seminumerical Algorithms, Ed. 2*; Addison Wesley, Reading, MA, 1981.

Kozen, D. and Landau, S., "Polynomial decomposition algorithms," *J. Symbolic Comp.* **7**/5, pp. 445–456 (1989).

Kronecker, L., "Grundzüge einer arithmetischen Theorie der algebraischen Grössen," *J. reine angew. Math.* **92**, pp. 1–122 (1882).

Landau, S., "Factoring polynomials over algebraic number fields," *SIAM J. Comp.* **14**, pp. 184–195 (1985). Erratum: *SIAM J. Comput.* **20**/5, p. 998 (1991).

Landau, S., "Simplification of nested radicals," *Proc. 30th Annual Symp. Foundations of Comp. Sci.*, pp. 314–319 (1989).

Landau, S. and Miller, G. L., "Solvability by radicals," *J. Comp. System Sci.* **30**, pp. 179–208 (1985).

Lenstra, A. K., "Factoring polynomials over algebraic number fields," *Proc. EUROCAL '83, Springer Lec. Notes Comp. Sci.* **162**, pp. 245–254 (1983).

Lenstra, A. K., "Factoring multivariate polynomials over algebraic number fields," *SIAM J. Comp.* **16**, pp. 591–598 (1987).

Lenstra, A. K., Lenstra, H. W., and Lovász, L., "Factoring polynomials with rational coefficients," *Math. Ann.* **261**, pp. 515–534 (1982).

Lickteig, T. M., "On semialgebraic decision complexity," *Tech. Report* **TR-90-052**, Internat. Computer Sci. Inst., Berkeley, California, September 1990. Habilitationsschrift.

Lidl, R. and Niederreiter, H., *Finite Fields*; Addison-Wesley, Reading, MA, 1983.

Linwood, D. A., "Roots of a polynomial via a parallel Newton's method," *Manuscript*, Dept. Math., California State University, Fresno, CA, July 1990.

Lipton, R. and Stockmeyer, L., "Evaluations of polynomials with superpreconditioning," *Proc. 8th ACM Symp. Theory Comp.*, pp. 174–180 (1976).

Lombardi, H., "Algèbre élémentaire en temps polynomial," *Thèse Doctorat*, Université de Franche-Comté, Besançon, France, June 1989. In French.

Loos, R., "Computing in algebraic extensions," in *Computer Algebra, 2nd ed.*, edited by B. Buchberger et al.; Springer Verlag, Vienna, pp. 173–187, 1982.

Lucks, M., "A fast implementation of polynomial factorization," *Proc. 1986 ACM Symp. Symbolic Algebraic Comp.*, pp. 228–232 (1986).

Marden, M., *The geometry of the zeros of a polynomial in a complex variable*; Math. Surveys **3**; AMS, Providence, R.I., 1949.

Melenk, H., Möller, H. M., and Neun, W., "On Gröbner bases computation on a supercomputer using REDUCE," *Tech. Report* **SC 88-2**, K. Zuse Zentrum Berlin, January 1988.

Menezes, A. J., van Oorschot, P. C., and Vanstone, S. A., "Some computational aspects of root finding in GF(q^m)," *Proc. ISSAC '88, Springer Lec. Notes Comput. Sci.* **358**, pp. 259–270 (1988).

Moenck, R. T., "On the efficiency of algorithms for polynomial factoring," *Math. Comp.* **31**, pp. 235–250 (1977).

Monagan, M. B., "A heuristic irreducibility test for univariate polynomials," *J. Symbolic Comput.* **13**/1, pp. 47–57 (1992).

Moore, P. M. A. and Norman, A. C., "Implementing a polynomial factorization problem," *Proc. 1981 ACM Symp. Symbolic Algebraic Comp.*, pp. 109–116 (1981).

Musser, D. R., "Multivariate polynomial factorization," *J. ACM* **22**, pp. 291–308 (1975).

Neff, C. A., "Specified precision polynomial root isolation is in NC," *Proc. 31st Annual Symp. Foundations Computer Sci.*, pp. 152–162 (1990).

Noether, E., "Ein algebraisches Kriterium für absolute Irreduzibilität," *Math. Ann.* **85**, pp. 26–33 (1922).

Pasquini, L. and Trigiante, D., "A globally convergent method for simultaneously finding polynomial roots," *Math. Comput.* **44**, pp. 135–149 (1985).

Pinkert, J. R., "An exact method for finding roots of a complex polynomial," *ACM Trans. Math. Software* **2**/4, pp. 351–363 (1976).

Plaisted, D. A., "Sparse complex polynomials and polynomial reducibility," *J. Comp. System Sci.* **14**, pp. 210–221 (1977).

Rabin, M. O., "Probabilistic algorithms in finite fields," *SIAM J. Comp.* **9**, pp. 273–280 (1980).

Rónyai, L., "Galois groups and factoring polynomials over finite fields," *Proc. 30th Annual Symp. Foundations of Comp. Sci.*, pp. 99–104 (1989).

Schmidt, W. M., *Equations over finite fields. An elementary approach*; Springer Lect. Notes Math. **536**; Springer Verlag, New York, N. Y., 1976.

Schoof, R. J., "Elliptic curves over finite fields and the computation of square roots mod p," *Math. Comp.* **44**, pp. 483–494 (1985).

Schwarz, Š., "On the reducibility of polynomials over a finite field," *Quart. J. Math. Oxford Ser. (2)* **7**, pp. 110–124 (1956).

Schönhage, A., "The fundamental theorem of algebra in terms of computational complexity," *Tech. Report*, Univ. Tübingen, 1982.

Schönhage, A., "Factorization of univariate integer polynomials by diophantine approximation and an improved basis reduction algorithm," *Proc. ICALP '84, Springer Lec. Notes Comp. Sci.* **172**, pp. 436–447 (1984).

Shoup, V., "New algorithms for finding irreducible polynomials over finite fields," *Math. Comput.* **54**/189, pp. 435–447 (1990a).

Shoup, V., "On the deterministic complexity of factoring polynomials over finite fields," *Inform. Process. Letters* **33**, pp. 261–267 (1990b).

Shoup, V., "Smoothness and factoring polynomials over finite fields," *Inform. Process. Letters* **38**, pp. 39–42 (1991a).

Shoup, V., "A fast deterministic algorithm for factoring polynomials over finite fields of small characteristic," in *Proc. 1991 Internat. Symp. Symbolic Algebraic Comput.*, edited by S. M. Watt; ACM Press, pp. 14–21, 1991b.

Shparlinskiy, I. E., "On some problems of theory of finite fields," *Manuscript*, Moscow, August 1990.

Smedley, T. J., "Fast methods for computation with algebraic numbers," *Ph. D. Thesis*, Dept. Comput. Sci., Univ. Waterloo, 1989.

Sprindžuk, V. G., "Arithmetic specializations in polynomials," *J. reine angew. Math.* **340**, pp. 26–52 (1983).

Strassen, V., "Algebraic complexity theory," in *Handbook of Theoretical Computer Science, Algorithms and Complexity (Volume A)*, edited by J. van Leeuwen; Elsevier Science Publ., Amsterdam, pp. 633–672, 1990.

Thiong ly, A., "A deterministic algorithm for factorizing polynomials over extensions GF(p^m), p a small prime," *J. Inform. Optim. Sci.* **10**, pp. 337-344 (1989).

Tonelli, A., "Bemerkung über die Auflösung quadratischer Congruenzen," *Nachrichten d. Akademie d. Wissenschaften in Göttingen*, pp. 344–346 (1891). In German.

Trager, B. M., "Algebraic factoring and rational function integration," *Proc. 1976 ACM Symp. Symbolic Algebraic Comp.*, pp. 219–228 (1976).

Trevison, V. and Wang, P., "Practical factorization of univariate polynomials over finite fields," in *Proc. 1991 Internat. Symp. Symbolic Algebraic Comput.*, edited by S. M. Watt; ACM Press, pp. 22–31, 1991.

Trotter, H. F., "Statistics on factoring polynomials mod p and p-adically," *SIGSAM Bulletin* **16**/3, pp. 24–29 (1982).

Viry, G., "Factorization of multivariate polynomials with coefficients in $\mathbf{F}_p$," *Manuscript*, Univ. Niamey, Niamey, Niger (West Africa), November 1990. *J. Symbolic Comput.*, to appear.

van der Waerden, B. L., "Eine Bemerkung über die Unzerlegbarkeit von Polynomen," *Math. Ann.* **102**, pp. 738–739 (1930). In German.

van der Waerden, B. L., *Moderne Algebra*; Springer Verlag, Berlin, 1940. English transl. publ. under the title "Modern algebra" by F. Ungar Publ. Co., New York, 1953

Wang, P. S., "An improved multivariate polynomial factorization algorithm," *Math. Comp.* **32**, pp. 1215–1231 (1978).

Wang, P. S., "Parallel univariate polynomial factorization on shared-memory multiprocessors," in *Proc. 1990 Internat. Symp. Symbolic Algebraic Comput.*, edited by S. Watanabe and M. Nagata; ACM Press, pp. 145–151, 1990.

Weinberger, P. J. and Rothschild, L. P., "Factoring polynomials over algebraic number fields," *ACM Trans. Math. Software* **2**, pp. 335–350 (1976).

Wiedemann, D., "Solving sparse linear equations over finite fields," *IEEE Trans. Inf. Theory* **IT-32**, pp. 54–62 (1986).

Wilf, H. S., "A global bisection algorithm for computing the zeros of polynomials in the complex plane," *J. ACM* **25**/3, pp. 415–420 (1978).

Yokoyama, K., Noro, M., and Takeshima, T., "On factoring multi-variate polynomials over algebraically closed fields," in *Proc. 1991 Internat. Symp. Symbolic Algebraic Comput.*, edited by S. M. Watt; ACM Press, p. 297, 1990.

Yun, D. Y. Y., "The Hensel lemma in algebraic manipulation," *Ph.D. Thesis*, M.I.T., 1974. Reprint: Garland Publ., New York 1980.

Zassenhaus, H., "On Hensel factorization I," *J. Number Theory* **1**, pp. 291–311 (1969).

Zassenhaus, H., "On Hensel factorization II," in *Instituto Nazionale di Alta Mat.*, Symposia Mathematica **15**; pp. 499-513, 1975.

Zassenhaus, H., "Polynomial time factoring of integral polynomials," *SIGSAM Bulletin* **15**/2, pp. 6–7 (1981).

Zippel, R., "Probabilistic algorithms for sparse polynomials," *Proc. EUROSAM '79, Springer Lec. Notes Comp. Sci.* **72**, pp. 216–226 (1979).

Zippel, R., "Newton's iteration and the sparse Hensel algorithm," *Proc. '81 ACM Symp. Symbolic Algebraic Comp.*, pp. 68–72 (1981).

Zippel, R., "Interpolating polynomials from their values," *J. Symbolic Comput.* **9**/3, pp. 375–403 (1990).

Zippel, R., "Rational function decomposition," in *Proc. 1991 Internat. Symp. Symbolic Algebraic Comput.*, edited by S. M. Watt; ACM Press, pp. 1–6, 1991.

Postscript

On November 21, 1991, around 6 o'clock in the morning, Prof. Hans Zassenhaus passed away. Hans Zassenhaus has fundamentally affected progress on the polynomial factoring problem: he is the inventor of Hensel-based algorithms; he considered randomizations as a means to speed factorization over large finite fields; he reduced the polynomial-time factorization problem over the rationals to a diophantine optimization problem solvable by lattice reduction; that reduction played a key role in my polynomial-time solution of the bivariate problem; and with David Cantor, he constructed a very space efficient factorization method over finite fields. I will miss Prof. Zassenhaus's thoughts.