Handbook of Finite Fields

by

Gary L. Mullen
Department of Mathematics
The Pennsylvania State University
University Park, PA 16802, U.S.A.
Email: mullen@math.psu.edu

and

Daniel Panario
School of Mathematics and Statistics
Carleton University
Ottawa, Ontario K1S 5B6, Canada
Email: daniel@math.carleton.ca

Contents

Part II: Theoretical Properties

II

Theoretical Properties

11

Algorithms

11.5 Factorization of multivariate polynomials

Erich Kaltofen, North Carolina State University
Gregoire Lecerf, CNRS & Ecole polytechnique

In this section we extend the univariate factorization techniques of the previous section to several variables. Two major ingredients are the reduction from the bivariate case to the univariate one, and the reduction from any number to two variables. We present most of the known techniques according to the representation of the input polynomial.

11.5.1 Factoring dense multivariate polynomials

11.5.1 Remark In this subsection we are concerned with different kinds of factorizations of a multivariate polynomial $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ stored in dense representation:

11.5.2 Definition Let R be any ring. A *dense representation* of a polynomial $f \in R[x_1, \ldots, x_n]$ is the data of the vector $(d_1, \ldots, d_n)$ of the partial degrees of f, and the vector of the coefficients of the monomials $x_1^{e_1} \cdots x_n^{e_n}$ for all $0 \le e_1 \le d_1, \ldots,$ $0 \le e_n \le d_n$, sorted in reverse lexicographical ordering on the exponents $(e_1, \ldots, e_n)$, which means that $(e_1, \ldots, e_n) < (e'_1, \ldots, e'_n)$ if, and only if, there exists j such that $(e_n = e'_n, \ldots, e_{j+1} = e'_{j+1}, \text{ and } e_j < e'_j)$.

11.5.3 Remark The representation of multivariate polynomials is an important issue, which has been discussed from the early ages of computer algebra [Czapor et al. 1992; Davenport et al. 1987; van der Hoeven and Lecerf 2010; Johnson 1974; Monagan and Pearce 2007, 2009, 2010; Stoutemyer 1984; Yan 1998].

11.5.1.1 Separable factorization

11.5.4 Remark Separable factorization can be seen as a preprocess to the other factorizations (squarefree, irreducible, and absolutely irreducible, as defined below), which allows to reduce to considering separable polynomials.

11.5.5 Definition Let R be an integral domain. A polynomial $f \in R[x]$ is *primitive* if the common divisors in R of all the coefficients of f are invertible in R.

11.5.6 Definition Let R be a unique factorization domain of characteristic p, and let E_p represent $\{1\}$ if $p = 0$ and $\{1, p, p^2, p^3, \ldots\}$ otherwise. If f is a primitive polynomial in $R[y]$ of degree $d \ge 1$, then the *separable decomposition* of f, written $\mathrm{Sep}(f)$, is defined to be the set $\mathrm{Sep}(f) := \{(f_1, q_1, m_1), \ldots, (f_s, q_s, m_s)\} \subseteq (R[y] \setminus R) \times E_p \times \mathbb{N}$, satisfying the following properties:

1. $f(y) = \prod_{i=1}^{s} f_i(y^{q_i})^{m_i}$,
2. for all $i \ne j$ in $\{1, \ldots, s\}$, $f_i(y^{q_i})$ and $f_j(y^{q_j})$ are coprime,
3. for all $i \in \{1, \ldots, s\}$, $m_i \pmod p \ne 0$,
4. for all $i \in \{1, \ldots, s\}$, f_i is separable and primitive,
5. for all $i \ne j$ in $\{1, \ldots, s\}$, $(q_i, m_i) \ne (q_j, m_j)$.

The process of computing the separable decomposition is the *separable factorization.*

11.5.7 Example With $R := \mathbb{F}_3$ and $f := y^2(y+1)^3(y+2)^4 = y^9 + 2y^8 + 2y^3 + y^2$, we have that $\mathrm{Sep}(f) = \{(y, 1, 2), (y + 1, 3, 1), (y + 2, 1, 4)\}$.

11.5.8 Example With $R := \mathbb{F}_3[x]$ and $f := (y + 2x)^7(y^3 + 2x)^3(y^6 + x)$, we have that $\mathrm{Sep}(f) = \{(y + 2x, 1, 7), (y + 2x^3, 9, 1), (y^2 + x, 3, 1)\}$.

11.5.9 Theorem [Mines et al. 1988, Chap. VI, Theorem 6.3] Any primitive polynomial $f \in R[y]$ admits a unique (up to permutations and units in R) separable decomposition, which only depends on the coefficients of f.

11.5.10 Remark Roughly speaking, the separable decomposition corresponds to sorting the roots of the given polynomial according to their multiplicity. A constructive proof of Theorem 11.5.9 can be found in [Mines et al. 1988, Chap. VI, Theorem 6.3], and another proof using the irreducible factorization in [Lecerf 2008, Proposition 4].

11.5.11 Remark Since the separable decomposition only depends on the coefficients of f it can be computed in any extension of R.

11.5.12 Theorem [Lecerf 2008, Proposition 5] If F is a field then the separable decomposition of a polynomial $f \in F[y]$ of degree d can be computed with $O(\mathsf{M}(d) \log d)$ arithmetic operations in F. Let us recall that $\mathsf{M}(d)$ represents a bound for the complexity of multiplying two polynomials of degree at most d with coefficients in a commutative ring with unity, in terms of the number of arithmetic operations in the latter ring.

11.5.13 Theorem [Lecerf 2008, Propositions 8 and 9] Let $R = F[x]$, where F is a field, and let $f \in F[x][y]$ be a primitive polynomial of degree d_x in x and d_y in y.

1. If F has cardinality at least $d_x(2d_y + 1) + 1$ then $\mathrm{Sep}(f)$ can be computed (deterministically) with $O(d_y(d_y\mathsf{M}(d_x) \log d_x + d_x\mathsf{M}(d_y) \log d_y))$ or $\tilde{O}(d_x d_y^2)$ operations in F.
2. If F has cardinality at least $4d_xd_y$ then $\mathrm{Sep}(f)$ can be computed with an *expected* number of $O(d_y\mathsf{M}(d_x) \log d_x + d_x\mathsf{M}(d_y) \log d_y))$ or $\tilde{O}(d_xd_y)$ operations in F.

Let us recall that $f(d) \in \tilde{O}(g(d))$ means that $f(d) \in g(d)(\log_2(3 + g(d)))^{O(1)}$. With the second randomized algorithm, the ouput is always correct, and the cost estimate is the average of the number of operations in F taken over all the possible executions.

11.5.1.2 Squarefree factorization

11.5.14 Definition If R is a unique factorization domain then the *squarefree decomposition* of $a \in R$, written $\mathrm{Sqr}(a)$, is the set of pairs (a_m, m), where a_m represents the product of all the irreducible factors of a of multiplicity m. The process of computing the squarefree decomposition is the *squarefree factorization.*

11.5.15 Definition For convenience, we say that a polynomial $f \in R[x_1, \ldots, x_n]$ is *primitive* (resp. *separable*) in x_i if it is so when seen in $R[x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n][x_i]$.

11.5.16 Algorithm ***Sketch of the algorithm squarefree factorization***

Input: a polynomial $f \in \mathbb{F}_q[x_1, \ldots, x_n]$, primitive in $x_1, \ldots, x_n$.

Output: the squarefree decomposition $\mathrm{Sqr}(f)$ of f.

1. First compute the separable decomposition of f seen in $\mathbb{F}_q[x_1,\ldots,x_{n-1}][x_n]$. Then for each separable factor g of $\mathrm{Sep}(f)$ compute the separable decomposition of g seen in $\mathbb{F}_q[x_1,\ldots,x_{n-2},x_n][x_{n-1}]$. Then for each separable factor h of $\mathrm{Sep}(g)$ compute the separable decomposition of h seen in $\mathbb{F}_q[x_1,\ldots,x_{n-3},x_{n-1},x_n][x_{n-2}]$, etc. At the end rewrite f as the product of polynomials of the form $f_i(x_1^{q_{i,1}},\ldots,x_n^{q_{i,n}})^{m_i}$, where the f_i are separable in $x_1,\ldots,x_n$, and where the $q_{i,j}$ are powers of p.
2. The squarefree factorization of each $f_i(x_1^{q_{i,1}},\ldots,x_n^{q_{i,n}})^{m_i}$ is simply obtained by extracting the $\min_{j\in\{1,\ldots,n\}} q_{i,j}$-th root of $f_i(x_1^{q_{i,1}},\ldots,x_n^{q_{i,n}})$.

11.5.17 Theorem [Lecerf 2008, Proposition 12] Let $f \in \mathbb{F}_q[x,y]$ be a polynomial of degree d_x in x and d_y in y. If $q \geq 4(3d_y+1)d_x$ then $\mathrm{Sqr}(f)$ can be computed with an *expected* number of $O(d_y\mathsf{M}(d_x)\log d_x + d_x\mathsf{M}(d_y)\log d_y))$ or $\tilde{O}(d_xd_y)$ operations in $\mathbb{F}_q$.

11.5.18 Remark Practical multivariate squarefree factorization algorithms have been designed in [Bernardin 1997] to be specifically efficient in small and medium sizes, when M does not behave as softly linear. Algorithms for deducing the squarefree decomposition from the separable one were proposed in [Gianni and Trager 1996] and then improved in [Lecerf 2008] in particular cases.

11.5.1.3 Bivariate irreducible factorization

11.5.19 Definition If R is a unique factorization domain then the *irreducible decomposition* of $a \in R$, written $\mathrm{Irr}(a)$, is the set of pairs (a_i, m_i), where a_i is an irreducible factor of a of multiplicity m_i. The process of computing the irreducible decomposition is the *irreducible factorization*.

11.5.20 Definition If F is a field then the *absolutely irreducible decomposition* of $f \in F[x_1,\ldots,x_n]$ is the irreducible decomposition of f in $\bar{F}[x_1,\ldots,x_n]$, where $\bar{F}$ represents the algebraic closure of F. The process of computing the absolutely irreducible decomposition is the *absolutely irreducible factorization*, or *absolute factorization.*

11.5.21 Remark In this section we will not discuss specific algorithms for computing the absolute factorization. In fact, whenever F is a finite field, the absolutely irreducible decomposition of $f \in F[x_1,\ldots,x_n]$ can be obtained from the irreducible decomposition over the algebraic extension of F of degree $\deg f$. For more details and advanced algorithms we refer the reader to [Chèze and Lecerf 2007].

See Also Absolute factorization intervenes for testing if a univariate rational function generates a permutation of a finite field as in the algorithms of [Kayal 2005; Ma and von zur Gathen 1995]. We refer the reader to Section **??**.

11.5.22 Theorem [Lecerf 2010, Theorem 2] Let $q = p^k$, and let $f \in \mathbb{F}_q[x,y]$ be a polynomial of degree d_x in x and d_y in y. If $q \geq 10d_xd_y$ then $\mathrm{Irr}(f)$ can be computed with factoring several polynomials in $\mathbb{F}_q[y]$ whose degree sum does not exceed $d_x + d_y$, plus an *expected* number of $\tilde{O}(k(d_xd_y)^{1.5})$ operations in $\mathbb{F}_p$.

11.5.23 Remark If q is not sufficiently large to apply Theorem 11.5.22 then one can compute the irreducible factorization of f over a slightly larger finite field, and then recover the factorization over $\mathbb{F}_q$ by computing the norm of the factors.

The algorithm underlying Theorem 11.5.22 summarizes as follows:

11.5.24 Algorithm ***Sketch of the lifting and recombination technique***

Input: a primitive and separable polynomial $f \in \mathbb{F}_q[x][y]$, of partial degrees d_x in x and d_y in y.

Output: the irreducible decomposition $\mathrm{Irr}(f)$ of f.

1. *Normalization.* If the cardinality of $\mathbb{F}_q$ is sufficiently large then a suitable shift of the variable x reduces the problem to the *normalized* case defined as follows:
$$\deg f(0,y) = d_y \text{ and } \mathrm{Res}\left(f(0,y), \frac{\partial f}{\partial y}(0,y)\right) \neq 0.$$
2. *Univariate factorization.* Compute $\mathrm{Irr}(f(0,y))$ in $\mathbb{F}_q[y]$.
3. *Lifting.* Use the classical *Hensel lifting* from the previously computed irreducible factors $\mathfrak{f}_1(0,y), \ldots, \mathfrak{f}_s(0,y)$ of $f(0,y)$ in order to deduce the irreducible *analytic decomposition* $\mathfrak{f}_1, \ldots, \mathfrak{f}_s$ of f in $\mathbb{F}_q[[x]][y]$ to a certain finite precision σ in x.
4. *Recombination.* Discover how the latter analytic factors $\mathfrak{f}_1, \ldots, \mathfrak{f}_s$ *recombine* into the irreducible factors.

11.5.25 Remark Since any proper factor g of f is the product of a subset of the analytic factors, the precision $\sigma = d_x$ is sufficient in Algorithm 11.5.24 to discover $\mathrm{Irr}(f)$ by means of exhaustive search. To be precise, it suffices to run over all the subsets S of $\{1, \ldots, s\}$ of cardinality at most $s/2$ and test whether the truncated polynomial of $\prod_{i \in S} \mathfrak{f}_i$ to precision d_x in $\mathbb{F}_q[x][y]$ divides f or not. This approach was originally popularized in computer algebra by Zassenhaus in [Zassenhaus 1969] in the context of factoring in $\mathbb{Q}[y]$ *via* the p-adic completion of $\mathbb{Q}$. The adaptation to two and several variables was first pioneered in [Musser 1975; Wang 1978; Wang and Rothschild 1975]. In particular, [Musser 1975] introduced coefficient field abstractions that marked the beginning of generic programming. Von zur Gathen adopted Musser's approach to valuation rings [von zur Gathen 1984]. The cost of this approach is, of course, exponential in s. However, as proved in [Gao and Lauder 2002] the cost of the recombination process behaves in softly linear time in average over finite fields, which explains the practical efficiency of this approach.

11.5.26 Remark For details concerning Hensel lifting, we refer the reader to [von zur Gathen and Gerhard 2003, Chap. 15], that contains a variant of the multifactor Hensel lifting first designed by Shoup for his C++ library NTL (`http://www.shoup.net`). An improvement obtained thanks to the transposition principle is proposed in [Bostan et al. 2004]. Parallelization has been studied in [Bernardin 1998].

11.5.27 Remark The first attempt to reduce the recombination stage to linear algebra seems to be due to Sasaki et al. [Sasaki et al. 1992; Sasaki and Sasaki 1993; Sasaki et al. 1991], with a method called the *trace recombination.* But the first successes in the design and proofs of complete algorithms are due to van Hoeij [van Hoeij 2002] for the factorization in $\mathbb{Z}[x]$, and then to Belabas et al. [Belabas et al. 2009] for $F(x)[y]$, with the *logarithmic derivative recombination* method, where the precision $\sigma = \deg f(\deg f - 1) + 1$ is shown to be sufficient in general. Then a precision linear in $\deg f$ in characteristic 0 or large enough characteristic has been shown to suffice in [Bostan et al. 2004; Lecerf 2006].

11.5.28 Remark In [Gao 2003], Gao designed the first softly quadratic time probabilistic reduction of the factorization problem from two to one variable whenever the characteristic of the coefficient field is zero or sufficiently large. His algorithm makes use of the first algebraic de Rham cohomology group of $F[x, y, 1/f(x, y)]$, as previously used by Ruppert [Ruppert 1986, 1999] for testing the absolute irreducibility. In fact, if f factors into $f_1 \cdots f_r$ over the algebraic closure of F then

$$\left(\frac{\hat{f}_i \frac{\partial f_i}{\partial x}}{f} dx + \frac{\hat{f}_i \frac{\partial f_i}{\partial y}}{f} dy \right)_{i \in \{1,\ldots,r\}}$$

is a basis of the latter group, where $\hat{f}_i := f/f_i$ (see [Ruppert 1986, Satz 2]). In consequence, this group can be obtained by searching for closed differential 1-forms with denominators f and numerators of degrees at most $\deg f - 1$, which can be easily done by solving a linear system. A nice presentation of Ruppert's results is made in Schinzel's book [Schinzel 2000, Chapter 3]. The algorithm underlying Theorem 11.5.22 makes use of these ideas in order to show that a precision $\sigma = d_x + 1$ of the series in the Hensel lifting suffices.

11.5.1.4 Reduction from any number to two variables

11.5.29 Remark Let $f \in F[x_1, \ldots, x_n]$ continue to denote a polynomial in n variables over a field F of total degree d. For any points $(\alpha_1, \ldots, \alpha_n)$, $(\beta_1, \ldots, \beta_n)$ and $(\gamma_1, \ldots, \gamma_n)$ in F^n, we define the bivariate polynomial $f_{\alpha,\beta,\gamma}$ in the variables x and y by $f_{\alpha,\beta,\gamma} := f(\alpha_1 x + \beta_1 y + \gamma_1, \ldots, \alpha_n x + \beta_n y + \gamma_n)$.

11.5.30 Theorem (Bertini's theorem, (*e.g.* [Shafarevich 1994, Chapter II, Section 6.1]) If f is irreducible, then there exists a proper Zariski open subset of $(F^n)^3$ such that $f_{\alpha,\beta,\gamma}$ is irreducible for any triple $(\alpha_1, \ldots, \alpha_n), (\beta_1, \ldots, \beta_n), (\gamma_1, \ldots, \gamma_n)$ in this subset.

11.5.31 Definition We say that, for any irreducible factor g of f, a triple $(\alpha_1, \ldots, \alpha_n), (\beta_1, \ldots, \beta_n), (\gamma_1, \ldots, \gamma_n)$ in $(F^n)^3$ is a *Bertinian good point* for g if $g(\alpha_1 x + \beta_1 y + \gamma_1, \ldots, \alpha_n x + \beta_n y + \gamma_n)$ is irreducible with the same total degree as g. In other words, the irreducible factors of f are in one-to-one correspondence with those of $f_{\alpha,\beta,\gamma}$. The complementary set of Bertinian good points is written $\mathcal{B}(f)$ and is the set of *Bertinian bad points.*

11.5.32 Remark For algorithmic purposes, the entries of $(\alpha_1, \ldots, \alpha_n)$, $(\beta_1, \ldots, \beta_n)$ and $(\gamma_1, \ldots, \gamma_n)$ must be taken in a finite subset S of F, so that we are naturally interested in upper bounding the number of Bertinian bad points in $(S^n)^3$. We refer to such a bound as a *quantitative Bertini theorem.* The density of Bertinian bad points with entries in a non-empty finite subset S of F is

$$\mathcal{B}(f, S) := \frac{|\mathcal{B}(f) \cap (S^n)^3|}{|S|^{3n}},$$

where $|S|$ represents the cardinality of S.

11.5.33 Theorem (Quantitative Bertini theorem [Kaltofen 1995, Corollary 2] and [Lecerf 2007, Corollary 8]) If F is a perfect field of characteristic p, and according to the above notation, we have that:

1. $\mathcal{B}(f, S) \leq (3d(d-1)+1)/|S|$ if $p \geq d(d-1)+1$,
2. $\mathcal{B}(f, S) \leq 2d^4/|S|$ otherwise.

11.5.34 Remark What we call "Bertini's theorem" here is a particular but central case of more general theorems such as in [Shafarevich 1994, Chapter II, Section 6.1]. As pointed out by Kaltofen [Kaltofen 1995], the special application of Bertini's theorem to reduce the factorization problem from several to two variables was already known by Hilbert [Hilbert 1892, p. 117]. This is why Kaltofen and some authors say "(effective) Hilbert Irreducibility Theorem" instead of "Bertini's theorem". For more historical details about Bertini's work, we refer the reader to [Jouanolou 1983; Kleiman 1998].

11.5.35 Remark Bertini's theorem was introduced in complexity theory by Heintz and Sieveking [Heintz and Sieveking 1981], and Kaltofen [Kaltofen 1982a]. It quickly became a cornerstone of many randomized factorization or reduction techniques including [von zur Gathen 1985; von zur Gathen and Kaltofen 1985; Kaltofen 1985a,b,c]. Over the field of complex numbers, Bajaj et al. [Bajaj et al. 1993] obtained the bound $\mathcal{B}(f,S) \leq (d^4 - 2d^3 + d^2 + d + 1)/|S|$ by following Mumford's proof [Mumford 1995, Theorem 4.17] of Bertini's theorem. Gao [Gao 2003] proved the bound $\mathcal{B}(f,S) \leq 2d^3/|S|$ whenever F has characteristic 0 or larger than $2d^2$. Then Chèze pointed out [Chèze 2004, Chapter 1] that the latter bound can be refined to $\mathcal{B}(f,S) \leq d(d^2-1)/|S|$ by using directly [Ruppert 1986, Satz C]. The paper [von zur Gathen 1985] contains a version for non-perfect fields with a bound that is exponential in d. If the cardinality $|F|$ is too small, one can switch to an extension (see Remark 11.5.64 below).

11.5.36 Corollary Let $\mathsf{S}(n,d)$ represent a cost function for the product of two power series over a field F in n variables truncated to precision d. Let $f \in \mathbb{F}_q[x_1,\ldots,x_n]$ be a polynomial of total degree d. If $q \geq 4d^4$ then $\mathrm{Irr}(F)$ can be computed with an *expected* number of $O(1)$ factorizations of polynomials in $\mathbb{F}_q[x,y]$ of total degree d, plus an *expected* number of $\tilde{O}(d\mathsf{S}(n-1,d))$ operations in $\mathbb{F}_q$.

11.5.37 Remark Softly optimal series products exist in particular cases [van der Hoeven and Lecerf 2010], for which the factorization thus reduces to the univariate case in expected softly linear time as soon as $n \geq 3$.

11.5.38 Remark The first deterministic polynomial time multivariate factorization algorithms are due to Kaltofen [Kaltofen 1982a,b]. Kaltofen constructed polynomial-time reductions to bi- (in 1981) and univariate (in 1982) factorization over an abstract field, which were discovered independently of the 1982 univariate factorization algorithm over the rationals by A. K. Lenstra, H. W. Lenstra, and Lovász [Lenstra et al. 1982]. Kaltofen's reduction to univariate factorization, however, was inspired by Zassenhaus's algorithm [Zassenhaus 1981]. For more references to work by others (Chistov, von zur Gathen, Grigoriev, A. K. Lenstra) that immediately followed, we refer the reader to Kaltofen's surveys [Kaltofen 1990, 1992, 2003], and to [von zur Gathen and Gerhard 2003].

11.5.39 Remark Polynomial factorization over finite fields has been implemented in Maple by Bernardin and Monagan [Bernardin and Monagan 1997]. Other practical techniques have been reported in [Noro and Yokoyama 2002]. At the present time, the most general algorithm is due to Steel [Steel 2005]: it handles all coefficient fields being explicitly finitely generated over their prime field, and it has been implemented within the Magma computer algebra system [Bosma et al. 1997]. Steel's algorithm actually completes and improves a previous approach investigated by Davenport and Trager [Davenport and Trager 1981].

11.5.2 Deterministic algorithms

It is possible via the rank of the Petr matrix or the distinct degree factorization algorithm to count the number of irreducible factors of a polynomial over a field $\mathbb{F}_q$ of characteristic p in deterministic polynomial time in $\log p$. The same remains true for multivariate polynomials [Kaltofen 1987; Gao et al. 2004], but the algorithms are not straightforward. In [Gao et al. 2004] a multivariate deterministic distinct degree factorization is presented. There "distinct degree" is with respect to any degree order.

11.5.3 Factoring sparse multivariate polynomials

11.5.40 Remark Let F be a field. A polynomial f in $F[x_1, \ldots, x_n]$ is made of a sum of terms, with each term composed of a coefficient and an exponent seen as a vector in $\mathbb{N}^n$. For any $e = (e_1, \ldots, e_n) \in \mathbb{N}^n$, we let f_e denote the coefficient of the monomial $x_1^{e_1} \cdots x_n^{e_n}$ in f. If a polynomial has only a few of nonzero terms in its dense representation, one prefers to use the following representation.

11.5.41 Definition A *sparse representation* of a multivariate polynomial stores the sequence of the nonzero terms as pairs of monomials and coefficients, sorted for instance in reverse lexicographical order.

11.5.42 Definition The *support* of f is $\mathrm{Supp}(f) := \{e \in \mathbb{N}^n \mid f_e \neq 0\}$.

11.5.3.1 Ostrowski's theorem

11.5.43 Definition The *Minkowski sum* of two subsets Q and R of $\mathbb{R}^n$, written $Q + R$, is $Q + R := \{e + f \mid (e, f) \in Q \times R\}$.

11.5.44 Definition A polytope in $\mathbb{R}^n$ is *integral* if all of its vertices are in $\mathbb{Z}^n$. An integral polytope P is said to be *integrally decomposable* if there exists two integral polytopes Q and R such that $P = Q+R$, where both Q and R have at least two points. Otherwise, P is *integrally indecomposable*.

11.5.45 Definition The *Newton polytope* of f, written $N(f)$, is the convex hull in $\mathbb{R}^n$ of $\mathrm{Supp}(f)$. The *integral convex hull* of f is the subset of points in $\mathbb{Z}^n$ lying in $N(f)$.

11.5.46 Theorem (Ostrowski's theorem [Ostrowski 1921], translated in [Ostrowski 1999]) If f factors into gh then we have $N(f) = N(g) + N(h)$.

11.5.3.2 Irreducibility tests based on indecomposability of polytopes

11.5.47 Corollary (*Irreducibility criterion*) [Gao 2001, p. 507] If $f \in F[x_1, \ldots, x_n]$ is a nonzero polynomial not divisible by any x_i, and if $N(f)$ is integrally indecomposable, then f is irreducible over any algebraic extension of F.

11.5.48 Theorem [Gao 2001, Theorem 4.2] Let P be an integral polytope in $\mathbb{R}^n$ contained in a hyperplane H and let $e \in \mathbb{Z}^n$ be a point lying outside of H. If $e_1, \ldots, e_k$ are all the vertices of P, then the convex hull of P and e is integrally indecomposable if, and only if, all the entries of $e - e_1$, $e - e_2$, $\ldots$, $e - e_k$ are coprime.

11.5.49 Theorem [Gao 2001, Theorem 4.11] Let P be an indecomposable integral polytope in $\mathbb{R}^n$ with at least two points, that is contained in a hyperplane H, and let $e \in \mathbb{R}^n$ be a point outside of H. Let S be any subset of points in $\mathbb{Z}^n$ contained in the convex hull of e and P. Then the convex hull of S and Q is integrally indecomposable.

See Also Irreducible multivariate polynomials are treated in Section **??**.

11.5.3.3 Sparse bivariate Hensel lifting driven by polytopes

11.5.50 Remark Let $f \in \mathbb{F}_p[x, y]$ be a polynomial with t nonzero terms and of total degree d such that $t < d$. Let r be a vector in $\mathbb{R}^2$, and let Γ be a subset of edges of $N(f)$ satisfying the following properties:

1. $N(f) \subseteq \Gamma + r\mathbb{R}_{\geq 0}$,
2. each of the two infinite edges of $\Gamma + r\mathbb{R}_{\geq 0}$ contains exactly one point of $N(f)$,
3. no proper subset of Γ satisfies the previous two conditions.

Assume furthermore that:

1. f factorizes into $f = gh$ for two proper factors g and h in $\mathbb{F}_p[x, y]$ with t_g and t_h terms respectively, such that $\max(t_g, t_h) \leq t^\lambda$ for some constant λ satisfying $1/2 \leq \lambda < 1$.
2. For each edge $\gamma \in \Gamma$ we are given polynomials g_γ and h_γ supported by γ_g and γ_h respectively, where γ_g and γ_h are the unique vertices or edges of $N(g)$ and $N(h)$ respectively such that $\gamma = \gamma_g + \gamma_h$.
3. For each edge $\gamma \in \Gamma$ the given polynomials g_γ and h_γ are coprime up to monomial factors.

11.5.51 Theorem [Abu Salem 2008, Theorem 28] Under the above assumptions, there exists an integral decomposition $N(f) = N(g)+N(h)$ such that $N(g)$ is not a single point or a line segment parallel to $r\mathbb{R}_{\geq 0}$. There exists at most one full factorization of f which extends the boundary factorization defined by the given $(g_\gamma)_{\gamma\in\Gamma}$ and $(h_\gamma)_{\gamma\in\Gamma}$. Assuming that d and p fit a machine word, this factorization can be computed, or shown not to exist, using $O(t^\lambda d^2 + t^{2\lambda} d \log d \log\log d + t^{4\lambda} d)$ bit-operations, and $O(t^\lambda d)$ bits of memory.

11.5.52 Remark Theorem 11.5.51 extends previous results from [Abu Salem et al. 2004]. Although it does not provide a complete factoring algorithm, it proves to be very efficient in practice for large particular problems.

11.5.3.4 Convex-dense bivariate factorization

11.5.53 Remark In the worst case, the size of the irreducible factorization is exponential in the sparse size of the polynomial f to be factored. However Theorem 11.5.46 ensures that the size of the output is upper bounded by the number π, called the *convex size*, of points in $\mathbb{Z}^n$ lying inside of $N(f)$. The next theorem to be presented reduces the bivariate sparse factorization to the usual dense case.

11.5.54 Definition The affine group over $\mathbb{Z}^2$, written $\mathrm{Aff}(\mathbb{Z}^2)$, is the set of the maps U

$$U : (i, j) \mapsto \begin{pmatrix} \alpha & \beta \\ \alpha' & \beta' \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} + \begin{pmatrix} \gamma \\ \gamma' \end{pmatrix}, \qquad (11.1)$$

with α, β, γ, α', β', and γ' in $\mathbb{Z}$, such that $\alpha\beta' - \alpha'\beta = \pm 1$.

11.5.55 Definition Let S be a finite subset of $\mathbb{Z}^2$. The set S is said to be *normalized* if it belongs to $\mathbb{N}^2$ and if it contains at least one point in $\{0\} \times \mathbb{N}$, and also at least one point in $\mathbb{N} \times \{0\}$.

11.5.56 Theorem [Berthomieu and Lecerf 2010, Theorem 1.2] For any normalized finite subset S of $\mathbb{Z}^2$, of cardinality σ, convex size π, and included in $[0, d_x] \times [0, d_y]$, one can compute an affine map $U \in \mathrm{Aff}(\mathbb{Z}^2)$ as in (11.1), together with $U(S)$, with $O(\sigma \log^2((d_x + 1)(d_y + 1)))$ bit-operations, such that $U(S)$ is normalized and contained in a block $[0, d'_x] \times [0, d'_y]$ satisfying $(d'_x + 1)(d'_y + 1) \leq 9\pi$.

11.5.57 Lemma For any field F, for any $f \in F[x, y]$ not divisible by x and y, for any U as in (11.1), the polynomial

$$U(f) := \sum_{(e_x,e_y)\in \mathrm{Supp}(f)} f_{(e_x,e_y)} x^{\alpha e_x+\beta e_y+\gamma} y^{\alpha' e_x+\beta' e_y+\gamma'}$$

is irreducible in $F[x, y, x^{-1}, y^{-1}]$ if, and only if, f is irreducible.

11.5.58 Remark In order to compute the irreducible factorization of F, we can compute a reduction map U as in Theorem 11.5.56 for $\mathrm{Supp}(f)$, then compute the irreducible factorization of $U(f)$, and finally apply U^{-1} to each factor. In this way we benefit from complexity bounds that only depend on the convex size π of f instead of its dense size $(d_x + 1)(d_y + 1)$.

11.5.4 Factoring straight-line programs and black boxes

11.5.59 Remark The sparse representation (see Definition 11.5.41) of a polynomial allows for space efficient storage of polynomials of very high degree, since the degree of the term $x^{2^{500}}$ can be represented by a 501 bit integer. Polynomials f whose sparse representation occupies $(\log(\deg f))^{O(1)}$ bit space are called *supersparse* (*lacunary*) [Lenstra 1999; Kaltofen and Koiran 2006]. While computing small degree factors of such polynomials over the rational numbers can be accomplished in bit time that is polynomial in the input size, over finite fields such tasks are NP- or co-NP-hard [Kipnis and Shamir 1999]. Here we have a situation where factoring over the rational numbers is provably easier than factoring over a sufficiently large finite field.

In [Kaltofen and Koiran 2005] it is shown, by transferring the construction in [Plaisted 1984], that several other operations on univariate and bivariate supersparse polynomials over a sufficiently large finite field are NP- or co-NP-hard. For instance, in [Kaltofen and Koiran 2005] the following is proven:

11.5.60 Theorem Suppose we have a Monte Carlo polynomial-time irreducibility test for supersparse polynomials in $\mathbb{F}_{2^m}[X, Y]$ for sufficiently large m. Then large integers can be factored in Las Vegas polynomial-time.

11.5.61 Remark A polynomial in n variables of (total) degree d can have $\binom{n+d}{n}$ terms, i.e., exponentially many terms in the number of variables. Sparse polynomials are those that have $(n + d)^{O(1)}$ non-zero terms. Note that, as in all asymptotic analysis, one considers not a single polynomial but an infinite set of sparse input polynomials that a given algorithm processes, now in polynomial time in the sparse size. By using the factorization $x^d - 1 = (x - 1)(x^{d-1} + \cdots + 1)$ one can easily generate examples where the sparse size of one irreducible factor is super-polynomially larger than the input size [von zur Gather and Kaltofen 1985, Example 5.1]. Motivated by algebraic computation models, straight-line programs were adopted as an alternate polynomial representation, first only for inputs [von zur Gathen 1985], but ultimately and importantly as a representation of the irreducible factors themselves [Kaltofen 1986, 1989]. Here is an example of a division-free straight-line program (single assignment program), where $\mathbb{F}$ is the field generated by those operands $c_1, c_2, \ldots$ which are constants, while $x_1, x_2, \ldots$ are input variables:

$$\begin{aligned} v_1 &\leftarrow c_1 \times x_1; \\ v_2 &\leftarrow x_2 - c_2; \\ v_3 &\leftarrow v_2 \times v_2; \\ v_4 &\leftarrow v_3 + v_1; \\ v_5 &\leftarrow v_4 \times x_3; \\ &\vdots \\ v_{101} &\leftarrow v_{100} + v_{51}; \end{aligned}$$

The variable v_{101} represents a polynomial in $\mathbb{F}[x_1, x_2, \ldots]$, which can be evaluated by use of the straight-line program. For instance, the determinant of an $n \times n$ matrix whose entries are n^2 variables can be represented, via Gaussian elimination, by a straight-line program with divisions of length $O(n^3)$. Because those divisions can cause divisions by zero on evaluation at certain points, it is desirable to remove them from such programs [Strassen 1973]: the shortest division-free straight-line program for the determinant that is known today has length $O(n^{2.7})$ and uses no constants other than 1 and -1 in $\mathbb{F}$ [Kaltofen and Villard 2004]. In any case, divisions can be removed by increasing the length by a factor $O((\deg f)^{1+\epsilon})$ for any $\epsilon > 0$. The 1986 algorithm in [Kaltofen 1986, 1989] produces from a straight-line program of length l for a polynomial of degree d in Monte Carlo random polynomial-time straight-line programs for the irreducible factors (and their multiplicities). The factor programs themselves have length $O(d^2 l + d^{3+\epsilon})$. Over finite fields of characteristic p, for an irreducible factor g of multiplicity $p^m m'$, where $\gcd(p, m') = 1$, a straight-line program for g^{p^m} is returned (see Remark 11.5.65 below). The algorithm is implemented in the Dagwood system [Freeman et al. 1988] and can factor matrix determinants. A shortcoming of the straight-line representations, which later were adopted by the TERA project, was exposed by the Dagwood program: the lengths, while polynomial in the input lengths, become quite large (over a million assignments). The construction, however, plays a key role in complexity theory [Kabanets and Impagliazzo 2004].

11.5.62 Remark Since polynomials represented by straight-line programs can be converted to sparse polynomials in polynomial-time in their sparse size by the algorithm in [Zippel 1979], the straight-line factorization algorithm brought to a successful conclusion the search for polynomial-time sparse factorizers. Previous attempts based on sparse Hensel lifting [von zur Gathen 1983; von zur Gather and Kaltofen 1985; Kaltofen 1985c; Zippel 1979, 1981], retained an exponential substep for many factors, namely the computation of the so-called right-side Hensel correction coefficients. The problem of computing the coefficient of a given term in a sparse product is in general #P-hard. Nonetheless, if a polynomial has only a few sparse factors, such sparse lifting can be quite efficient, in practice.

11.5.63 Remark Instead of straight-line programs, one can use a full-fledged programmed procedure that evaluates the input polynomial. The irreducible factors are then evaluated at values for the variables by another procedure that makes ("oracle") calls to the input evaluation procedure. Thus is the genesis of algorithms for *black box* polynomials [Kaltofen and Trager 1988, 1990].

The idea is the following: Suppose one can call a black evaluation box for the polynomial $f(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$. First, uniformly randomly select from a sufficiently large finite set field elements a_i, c_i $(2 \leq i \leq n)$ and b_j $(1 \leq j \leq n)$ and interpolate and factor the bivariate image

$$\hat{f}(X, Y) = f(X + b_1, c_2 Y + a_2 X + b_2, \ldots, c_n Y + a_n X + b_n) = \prod_{k=1}^{r} \hat{g}_k(X, Y)^{e_k}.$$

By the effective Hilbert Irreducibility Theorems 11.5.33 above, the irreducible polynomials $\hat{g}_k$ are with high probability bivariate images of the irreducible factors $h_k(x_1, \ldots, x_n)$ of f. For small coefficient fields we shall assume that the black box can evaluate f at elements in a finite algebraic extension E of $\mathbb{F}$. Already the bivariate interpolation algorithm may require such an extension in order to have sufficiently many distinct points.

11.5.64 Remark If one selects an extension E of degree $[E : \mathbb{F}] > \deg(f)$ that is a prime number, all h_k remain irreducible over that extension. Indeed, the Frobenius norm $\mathrm{Norm}_{E/\mathbb{F}}(\tilde{h}) \in \mathbb{F}[x_1, \ldots, x_n]$ of a possible non-trivial irreducible factor $\tilde{h} \in E[x_1, \ldots, x_n]$ of an h_k must be a power of an irreducible polynomial over $\mathbb{F}$, hence a power of h_k itself. For otherwise $\gcd(h_k, \mathrm{Norm}_{E/\mathbb{F}}(\tilde{h}))$ would constitute a non-trivial factor of h_k over $\mathbb{F}$. But then $\deg(\tilde{h}) \cdot [E : \mathbb{F}] = \deg(h_k) \cdot m$, where m is the exponent of that power, and because $[E : \mathbb{F}]$ is a prime $> \deg(f) \geq \deg(h_k)$, we obtain the contradiction $\deg(\tilde{h}) = \deg(h_k) \cdot (m/[E : \mathbb{F}]) \geq \deg(h_k)$.

Remark 11.5.63 continued. Now the black box for evaluating all $h_k(\xi_1, \ldots, \xi_n)$ at field elements $\xi_i \in \mathbb{F}$ stores ("hard-wires") the a_i, b_j and the factors $g_k(X) = \hat{g}_k(X, 0)$ in its constant pool. We note that g_k are not necessarily irreducible, but with high probability they are pairwise relatively prime [Kaltofen and Trager 1990, Section 2, Step 3], and their leading terms only depend on the variable X. The black box first interpolates

$$\begin{aligned}\bar{f}(X, Y) = f(X + b_1, Y(\xi_2 - a_2(\xi_1 - b_1) - b_2) + a_2 X + b_2,& \\ \ldots, Y(\xi_n - a_n(\xi_1 - b_1) - b_n) + a_n X + b_n)& \qquad (11.2)\end{aligned}$$

and then factors $\bar{f}$ such that

$$\bar{f}(X, Y) = \prod_{k=1}^{r} \bar{h}_k(X, Y)^{e_k} \quad \text{with} \quad \bar{h}_k(X, 0) = g_k(X). \qquad (11.3)$$

We note that again the $\bar{h}_k$ are not necessarily irreducible. One may Hensel-lift the factorization

$$f(X + b_1, a_2 X + b_2, \ldots, a_n X + b_n) = \prod_{k=1}^{r} g_k(X)^{e_k} \qquad (11.4)$$

provided none of the multiplicities e_k is divisible by p. Otherwise, one can fully factor $\bar{f}(X, Y)$ and lump (multiply) those irreducible factors $\bar{h}_\kappa(X, Y)$ together where $\bar{h}_\kappa(X, 0)$ divide one and the same $g_k(X)$. Alternatively, if p^m divides e_k one could lift the p^m-th power of g_k and take a p^m-th root of the lifted factor. We have $\bar{f}(\xi_1 - b_1, 1) = f(\xi_1, \ldots, \xi_n)$, and for all k we obtain $\bar{h}_k(\xi_1 - b_1, 1) = h_k(\xi_1, \ldots, \xi_n)$. We observe that the scalar multiple of h_k is fixed in all evaluations by the choice of g_k.

11.5.65 Remark Over finite coefficient fields, there is no restriction on the multiplicities e_k. One does not obtain a pure straight-line program for the polynomial h_k because a bivariate factorization of $\bar{f}$ or a p^m-th root of the lifted factor, which depend on the evaluation points ξ_i, are performed on each evaluation. One can obtain straight-line polynomials that equal the irreducible factors modulo $(x_1^q - x_1, \ldots, x_n^q - x_n)$ by powering by q/p^m, where $q < \infty$ is the cardinality of the coefficient field. Those straight-line programs produce correct evaluations of the irreducible factors.

11.5.66 Remark The blackbox factorization algorithm is implemented in the FoxBox system [Díaz and Kaltofen 1998]. The size blowup experienced in the straight-line factorization algorithm does not occur. In fact, the factor evaluation black box makes $O(\deg(f)^2)$ calls to the black box for f and factors a bivariate polynomial, either by lifting (11.4) or, if multiplicities are divisible by the characteristic, by factoring $\bar{f}$. The program is fixed except for the constants a_i, b_j and the polynomials g_k.

11.5.67 Remark We conclude that the sparse representations of the factors can be recovered by sparse interpolation over a finite field (see [Kaltofen and Lee 2003] and the literature cited there). Dense factors can be identified to have more than a given number of terms and skipped.

Bibliography

Abu Salem, F., Gao, S., and Lauder, A. G. B. Factoring polynomials via polytopes. In *ISSAC '04: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, pages 4–11, New York, 2004. ACM Press. [10]

Abu Salem, Fatima K. An efficient sparse adaptation of the polytope method over $\mathbb{F}_p$ and a record-high binary bivariate factorisation. *J. Symbolic Comput.*, 43(5): 311–341, 2008. [10]

Bajaj, C., Canny, J., Garrity, T., and Warren, J. Factoring rational polynomials over the complex numbers. *SIAM J. Comput.*, 22(2):318–331, 1993. [8]

Belabas, K., van Hoeij, M., Klüners, J., and Steel, A. Factoring polynomials over global fields. *J. Théor. Nombres Bordeaux*, 21(1):15–39, 2009. [6]

Bernardin, L. On square-free factorization of multivariate polynomials over a finite field. *Theoret. Comput. Sci.*, 187(1-2):105–116, 1997. [5]

Bernardin, L. On bivariate Hensel lifting and its parallelization. In *ISSAC '98: Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, pages 96–100, New York, 1998. ACM Press. [6]

Bernardin, L. and Monagan, M. B. Efficient multivariate factorization over finite fields. In *Applied algebra, algebraic algorithms and error-correcting codes (Toulouse, 1997)*, volume 1255 of *Lecture Notes in Comput. Sci.*, pages 15–28. Springer-Verlag, 1997. [8]

Berthomieu, J. and Lecerf, G. Convex-dense bivariate polynomial factorization. Manuscript available from `http://hal.archives-ouvertes.fr/hal-00526659`, to appear in Math. Comp., 2010. [11]

Bosma, W., Cannon, J., and Playoust, C. The Magma algebra system I: The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. [8]

Bostan, A., Lecerf, G., Salvy, B., Schost, É., and Wiebelt, B. Complexity issues in bivariate polynomial factorization. In *ISSAC '04: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, pages 42–49, New York, 2004. ACM Press. [6]

Chèze, G. *Des méthodes symboliques-numériques et exactes pour la factorisation absolue des polynômes en deux variables.* PhD thesis, Université de Nice-Sophia Antipolis (France), 2004. [8]

Chèze, G. and Lecerf, G. Lifting and recombination techniques for absolute factorization. *J. Complexity*, 23(3):380–420, 2007. [5]

Czapor, S., Geddes, K., and Labahn, G. *Algorithms for Computer Algebra.* Kluwer Academic Publishers, 1992. [3]

Davenport, J. H., Siret, Y., and Tournier, É. *Calcul formel : systèmes et algorithmes de manipulations algébriques.* Masson, Paris, France, 1987. [3]

Davenport, J. H. and Trager, B. M. Factorization over finitely generated fields. In *SYMSAC'81: Proceedings of the fourth ACM symposium on Symbolic and algebraic computation*, pages 200–205. ACM Press, 1981. [8]

Díaz, A. and Kaltofen, E. FoxBox a system for manipulating symbolic objects in black box representation. In *ISSAC '98: Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, pages 30–37, 1998. [14]

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):218–240, 1988. [12]

Gao, S. Absolute irreducibility of polynomials via Newton polytopes. *J. Algebra*, 237 (2):501–520, 2001. [9, 10]

Gao, S. Factoring multivariate polynomials via partial differential equations. *Math. Comp.*, 72(242):801–822, 2003. [6, 8]

Gao, S., Kaltofen, E., and Lauder, A. Deterministic distinct degree factorization for polynomials over finite fields. *J. Symbolic Comput.*, 38(6):1461–1470, 2004. [9]

Gao, S. and Lauder, A. G. B. Hensel lifting and bivariate polynomial factorisation over finite fields. *Math. Comp.*, 71(240):1663–1676, 2002. [6]

von zur Gathen, J. Factoring sparse multivariate polynomials. In *24th Annual IEEE Symposium on Foundations of Computer Science*, pages 172–179, Los Alamitos, CA, USA, 1983. IEEE Computer Society. [12]

von zur Gathen, J. Hensel and Newton methods in valuation rings. *Math. Comp.*, 42 (166):637–661, 1984. [6]

von zur Gathen, J. Irreducibility of multivariate polynomials. *J. Comput. System Sci.*, 31(2):225–264, 1985. Special issue: Twenty-fourth annual symposium on the foundations of computer science (Tucson, Ariz., 1983). [8, 12]

von zur Gathen, J. and Kaltofen, E. Factoring multivariate polynomials over finite fields. *Math. Comp.*, 45:251–261, 1985. [8]

von zur Gather, J. and Kaltofen, E. Factoring sparse multivariate polynomials. *J. Comput. System Sci.*, 31:265–287, 1985. [12]

Gianni, P. and Trager, B. Square-free algorithms in positive characteristic. *Appl. Alg. Eng. Comm. Comp.*, 7(1):1–14, 1996. [5]

Heintz, J. and Sieveking, M. Absolute primality of polynomials is decidable in random polynomial time in the number of variables. In *Automata, languages and programming (Akko, 1981)*, volume 115 of *Lecture Notes in Comput. Sci.*, pages 16–28. Springer-Verlag, 1981. [8]

Hilbert, D. Ueber die Irreducibilität ganzer rationaler Functionen mit ganzzahligen Coefficienten. *J. Reine Angew. Math.*, 110, 1892. [8]

van Hoeij, M. Factoring polynomials and the knapsack problem. *J. Number Theory*, 95(2):167–189, 2002. [6]

van der Hoeven, J. and Lecerf, G. On the bit-complexity of sparse polynomial and series multiplication. Manuscript available from `http://hal.archives-ouvertes.fr/hal-00476223/fr`, 2010. [3, 8]

Johnson, S. C. Sparse polynomial arithmetic. *ACM SIGSAM Bull.*, 8(3):63–71, 1974. [3]

Jouanolou, J.-P. *Théorèmes de Bertini et applications*, volume 42 of *Progress in Mathematics*. Birkhäuser Boston, 1983. [8]

Kabanets, V. and Impagliazzo, R. Derandomizing polynomial identity tests means proving circuit lower bounds. *Comput complexity*, 13(1-2):1–46, 2004. [12]

Kaltofen, E. A polynomial reduction from multivariate to bivariate integral polynomial factorization. In *Proceedings of the 14th Symposium on Theory of Computing*, pages 261–266. ACM Press, 1982a. [8]

Kaltofen, E. A polynomial-time reduction from bivariate to univariate integral polynomial factorization. In *Proc. 23rd Annual Symp. Foundations of Comp. Sci.*, pages 57–64. IEEE, 1982b. [8]

Kaltofen, E. Effective Hilbert irreducibility. *Information and Control*, 66:123–137, 1985a. [8]

Kaltofen, E. Fast parallel absolute irreducibility testing. *J. Symbolic Comput.*, 1(1): 57–67, 1985b. [8]

Kaltofen, E. Sparse Hensel lifting. In *Proceedings of EUROCAL '85, Vol. 2 (Linz, 1985)*, volume 204 of *Lecture Notes in Comput. Sci.*, pages 4–17. Springer-Verlag, 1985c. [8, 12]

Kaltofen, E. Uniform closure properties of p-computable functions. In *Proc. 18th Annual ACM Symp. Theory Comput.*, pages 330–337. ACM, 1986. Also published as part of [Kaltofen 1988] and [Kaltofen 1989]. [12]

Kaltofen, E. Deterministic irreducibility testing of polynomials over large finite fields. *J. Symbolic Comput.*, 4:77–82, 1987. [9]

Kaltofen, E. Greatest common divisors of polynomials given by straight-line programs. *J. ACM*, 35(1):231–264, 1988. [17]

Kaltofen, E. Factorization of polynomials given by straight-line programs. In Micali, S., editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 375–412. JAI Press Inc., Greenwhich, Connecticut, 1989. [12, 17]

Kaltofen, E. Polynomial factorization 1982-1986. In Chudnovsky, D. V. and Jenks, R. D., editors, *Computers in Mathematics*, volume 125 of *Lecture Notes in Pure and Applied Mathematics*, pages 285–309. Marcel Dekker, New York, N. Y., 1990. [8]

Kaltofen, E. Polynomial factorization 1987-1991. In Simon, I., editor, *Proc. LATIN '92*, volume 583 of *Lect. Notes Comput. Sci.*, pages 294–313. Springer-Verlag, 1992. [8]

Kaltofen, E. Effective Noether irreducibility forms and applications. *J. Comput. System Sci.*, 50(2):274–295, 1995. [7, 8]

Kaltofen, E. Polynomial factorization: a success story. In *ISSAC '03: Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 3–4. ACM Press, 2003. [8]

Kaltofen, Erich and Koiran, Pascal. On the complexity of factoring bivariate supersparse (lacunary) polynomials. In *ISSAC '05: Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation*, pages 208–215, 2005. [11]

Kaltofen, E. and Koiran, P. Finding small degree factors of multivariate supersparse (lacunary) polynomials over algebraic number fields. In *ISSAC '06: Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation*, pages 162–168, 2006. [11]

Kaltofen, E. and Lee, Wen-shin. Early termination in sparse interpolation algorithms. *J. Symbolic Comput.*, 36(3–4):365–400, 2003. [14]

Kaltofen, E. and Trager, B. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. In *Proc. 29th Annual Symp. Foundations of Comp. Sci.*, pages 296–305. IEEE, 1988. [13]

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):301–320, 1990. [13]

Kaltofen, Erich and Villard, Gilles. On the complexity of computing determinants. *Comput. Complexity*, 13(3-4):91–130, 2004. ISSN 1016-3328. URL `http://dx.doi.org/10.1007/s00037-004-0185-3`. [12]

Kayal, Neeraj. Recognizing permutation functions in polynomial time. *ECCC*, TR05-008, 2005. [5]

Kipnis, A. and Shamir, A. Cryptanalysis of the HFE public key cryptosystem by relinearization. In Wiener, Michael J., editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 19–30. Springer-Verlag, 1999. [11]

Kleiman, S. L. Bertini and his two fundamental theorems. *Rend. Circ. Mat. Palermo (2) Suppl.*, 55:9–37, 1998. Studies in the history of modern mathematics, III. [8]

Lecerf, G. Sharp precision in Hensel lifting for bivariate polynomial factorization. *Math. Comp.*, 75:921–933, 2006. [6]

Lecerf, G. Improved dense multivariate polynomial factorization algorithms. *J. Symbolic Comput.*, 42(4):477–494, 2007. [7]

Lecerf, G. Fast separable factorization and applications. *Appl. Alg. Eng. Comm. Comp.*, 19(2), 2008. [4, 5]

Lecerf, G. New recombination algorithms for bivariate polynomial factorization based on Hensel lifting. *Appl. Alg. Eng. Comm. Comp.*, 21(2):151–176, 2010. [5]

Lenstra, A. K., Lenstra, Jr., H. W., and Lovász, L. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982. [8]

Lenstra, Jr., H. W. Finding small degree factors of lacunary polynomials. In Győry, Kálmán, Iwaniec, Henryk, and Urbanowicz, Jerzy, editors, *Number Theory in Progress*, volume 1 Diophantine Problems and Polynomials, pages 267–276. Stefan Banach Internat. Center, Walter de Gruyter Berlin/New York, 1999. ISBN 3-11-015715-2. Proc. Internat. Conf. Number Theory in Honor of the 60th Birthday of Andrzej Schinzel, Zakopane, Poland June 30–July 9, 1997. [11]

Ma, Keju and von zur Gathen, Joachim. The computational complexity of recognizing permutation functions. *Comput. Complexity*, 5(1):76–97, 1995. ISSN 1016-3328. [5]

Mines, R., Richman, F., and Ruitenburg, W. *A course in constructive algebra*. Universitext. Springer-Verlag, 1988. [4]

Monagan, M. and Pearce, R. Polynomial division using dynamic arrays, heaps, and packed exponent vectors. In *Proc. of CASC 2007*, pages 295–315. Springer-Verlag, 2007. [3]

Monagan, M. and Pearce, R. Parallel sparse polynomial multiplication using heaps. In *ISSAC '09: Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation*, pages 263–270, New York, NY, USA, 2009. ACM Press. [3]

Monagan, M. and Pearce, R. Sparse polynomial multiplication and division in Maple 14. *ACM Communications in Computer Algebra*, 44(3/4), 2010. [3]

Mumford, David. *Algebraic geometry. I*. Classics in Mathematics. Springer-Verlag, Berlin, 1995. ISBN 3-540-58657-1. Complex projective varieties, Reprint of the 1976 edition. [8]

Musser, D. R. Multivariate polynomial factorization. *J. Assoc. Comput. Mach.*, 22: 291–308, 1975. [6]

Noro, M. and Yokoyama, K. Yet another practical implementation of polynomial factorization over finite fields. In *ISSAC '02: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, pages 200–206. ACM Press, 2002. [8]

Ostrowski, A. M. Über die Bedeutung der Theorie der konvexen Polyeder für die formale Algebra. *Jahresber. Deutsch. Math.-Verein.*, 30(2):98–99, 1921. Talk given at *Der Deutsche Mathematikertag vom 18–24 September 1921 in Jena*. [9, 19]

Ostrowski, A. M. On the significance of the theory of convex polyhedra for formal algebra. *ACM SIGSAM Bull.*, 33(1):5, 1999. Translated from [Ostrowski 1921]. [9]

Plaisted, D. A. New NP-hard and NP-complete polynomial and integer divisibility problems. *Theoret. Comput. Sci.*, 13:125–138, 1984. [11]

Ruppert, W. M. Reduzibilität ebener Kurven. *J. Reine Angew. Math.*, 369:167–191, 1986. [7, 8]

Ruppert, W. M. Reducibility of polynomials $f(x, y)$ modulo p. *J. Number Theory*, 77 (1):62–70, 1999. [7]

Sasaki, T., Saito, T., and Hilano, T. Analysis of approximate factorization algorithm. I. *Japan J. Indust. Appl. Math.*, 9(3):351–368, 1992. [6]

Sasaki, T. and Sasaki, M. A unified method for multivariate polynomial factorizations. *Japan J. Indust. Appl. Math.*, 10(1):21–39, 1993. [6]

Sasaki, T., Suzuki, M., Kolář, M., and Sasaki, M. Approximate factorization of multivariate polynomials and absolute irreducibility testing. *Japan J. Indust. Appl. Math.*, 8(3):357–375, 1991. [6]

Schinzel, A. *Polynomials with special regard to reducibility*, volume 77 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2000. [7]

Shafarevich, I. R. *Basic algebraic geometry. 1 Varieties in projective space*. Springer-Verlag, second edition, 1994. ISBN 3-540-54812-2. [7, 8]

Steel, A. Conquering inseparability: primary decomposition and multivariate factorization over algebraic function fields of positive characteristic. *J. Symbolic Comput.*, 40(3):1053–1075, 2005. [8]

Stoutemyer, D. R. Which polynomial representation is best? In *Proceedings of the 1984 MACSYMA Users' Conference: Schenectady, New York, July 23–25, 1984*, pages 221–243, 1984. [3]

Strassen, V. Vermeidung von Divisionen. *J. Reine Angew. Math.*, 264:182–202, 1973. [12]

von zur Gathen, J. and Gerhard, J. *Modern computer algebra*. Cambridge University Press, Cambridge, second edition, 2003. ISBN 0-521-82646-2. [6, 8]

Wang, P. S. An improved multivariate polynomial factoring algorithm. *Math. Comp.*, 32(144):1215–1231, 1978. [6]

Wang, P. S. and Rothschild, L. P. Factoring multivariate polynomials over the integers. *Math. Comp.*, 29:935–950, 1975. [6]

Yan, T. The geobucket data structure for polynomials. *J. Symbolic Comput.*, 25(3): 285–293, 1998. [3]

Zassenhaus, H. On Hensel factorization I. *J. Number Theory*, 1(1):291–311, 1969. [6]

Zassenhaus, H. Polynomial time factoring of integral polynomials. *ACM SIGSAM Bull.*, 15(2):6–7, 1981. [8]

Zippel, R. Probabilistic algorithms for sparse polynomials. In *EUROSAM '79: Proceedings of the International Symposium on Symbolic and Algebraic Computation*, number 72 in Lecture Notes in Comput. Sci., pages 216–226. Springer-Verlag, 1979. [12]

Zippel, R. Newton's iteration and the sparse Hensel algorithm (Extended Abstract). In *SYMSAC '81: Proceedings of the fourth ACM Symposium on Symbolic and Algebraic Computation*, pages 68–72, New York, 1981. ACM Press. [12]

Index