Article Submitted to Journal of Symbolic Computation

Deterministic distinct-degree factorisation of polynomials over finite fields*

Shuhong Gao[1], Erich Kaltofen[2] and Alan G.B. Lauder[3]

[1] *Department of Mathematical Sciences*
Clemson University
Clemson, SC 29634-0975 USA
E-mail: sgao@math.clemson.edu

[2] *Department of Mathematics*
North Carolina State University
Raleigh, NC 27695-8205, USA
E-mail: kaltofen@math.ncsu.edu

[3] *Mathematical Institute*
Oxford University
Oxford OX1 3LB, UK
E-mail: lauder@maths.ox.ac.uk

Abstract

A deterministic polynomial time algorithm is presented for finding the distinct-degree factorization of multivariate polynomials over finite fields. As a consequence, one can count the number of irreducible factors of polynomials over finite fields in deterministic polynomial time, thus resolving an open problem of Kaltofen from 1987.

1. Introduction

It is a classical result in algorithmic number theory that one may compute the distinct-degree factorization of a univariate polynomial over a finite field in deterministic polynomial time. For a polynomial f of degree n this is a factorization of the form $f = \prod_{d=1}^{n} f^{[d]}$ where $f^{[d]}$ is the product of all irreducible factors of f of degree d. That this may be done follows easily from the theory of finite field extensions; von zur Gathen and Gerhard (1999, notes to section 14)

*S. Gao was supported in part by NSF under Grant DMS9970637, NSA under Grant MDA904-00-1-0048 and ONR under Grant N00014-00-1-0565. E. Kaltofen was supported in part by NSF under Grants CCR9988177, DMS9977392, and INT9726763. A. Lauder gratefully acknowledges the support of the Marr Educational Trust and Wolfson College, Oxford. This research was done while the authors were members at the Mathematical Sciences Research Institute in Berkeley, CA, USA.
Mathematics Subject Classification 2000: Primary 68W30; Secondary 11T06, 13P05. *Key words and phrases*: multivariate polynomial, deterministic algorithm, distinct-degree factorisation.

trace the standard algorithm back to C. F. Gauss. For polynomials in more than one variable no such analogous theory is available. Moreover, a straightforward approach based upon computing the distinct-degree factorization of some univariate image of a multivariate polynomial fails. We take a different approach and present a deterministic polynomial time algorithm for distinct-degree factorization of multivariate polynomials over finite fields, which does not use any univariate factorization subroutine. The algorithm is based upon earlier work of Kaltofen, who was able to show that one may test irreducibility of multivariate polynomials in deterministic polynomial time (Kaltofen 1987). In the same paper, Kaltofen asks whether it is also possible to count the number of irreducible factors; that this may be done is a consequence of our more general result.

Our paper is organised in the following way. Section 2 contains preliminary results on linear systems over polynomial algebras. In Section 3 we describe an algorithm for distinct-degree factorization, based upon the method of Kaltofen, which uses a randomised univariate factoring algorithm as a subroutine. Next in Section 4 we modify this algorithm using the methods of Section 2 so as to remove the need for any univariate factorization, just computing gcds instead. As in (Kaltofen 1987) for simplicity we shall focus on the bivariate case; the method extends to all multivariate polynomials, and we shall briefly discuss this at the end of the paper.

2. Linear systems over polynomial algebras

In this section we consider homogeneous linear systems over rings of the form

$$R_f := \mathbb{F}_q[z]/(f(z)) \cong \bigoplus_{j=1}^{r} \mathbb{F}_q[z]/(f_j(z))$$

where $f \in \mathbb{F}_q[z]$ is a squarefree polynomial of degree n with irreducible factors f_j. The Berlekamp subalgebra of R_f is

$$B_f := \{g \in R_f \mid g^q = g\}.$$

We have

$$B_f \cong \bigoplus_{j=1}^{r} B_{f_j}$$

and each $B_{f_j} \cong \mathbb{F}_q$. We shall denote by π_{f_j} the projection of R_f onto the jth summand $\mathbb{F}_q[z]/(f_j(z))$, and also the restriction of this map to B_f (which has codomain $B_{f_j} \cong \mathbb{F}_q$). Thus for $g \in R_f$,

$$\pi_{f_j}(g) := g \bmod f_j.$$

Let $\mathcal{L}$ be a linear system given by

$$Lv = 0 \tag{1}$$

where L is a matrix with entries in R_f. This is a linear system over R_f, however we wish to find solutions v to this system which have entries in B_f. Note the embeddings $\mathbb{F}_q \subseteq B_f \subseteq R_f$. The set of vectors with entries in B_f which are solutions to $\mathcal{L}$ forms a B_f-module and also an $\mathbb{F}_q$-vector space, under the usual vector addition and multiplication by scalars. It will not in general be a R_f-module, unless $B_f = R_f$. Note that not all sets of vectors with entries in B_f which form a $\mathbb{F}_q$-vector space necessarily form a B_f-module. For example, the ring $\mathbb{F}_q$ itself is a $\mathbb{F}_q$-vector space but not a B_f-module, unless $\mathbb{F}_q = B_f$, and the methods we discuss are not directly relevant to this situation.

LEMMA 2.1: *One may compute a basis over $\mathbb{F}_q$ of solutions to (1) with entries in B_f in deterministic polynomial time.*

Proof: Note that we have an explicit basis for R_f over $\mathbb{F}_q$, namely $1, z, \cdots, z^{n-1} \bmod f$ where n is the degree of f. Since qth power is a linear map of R_f over $\mathbb{F}_q$, a basis for B_f over $\mathbb{F}_q$ can be computed in deterministic polynomial time ((Butler 1954; Lidl and Niederreiter 1983); the idea goes back to Karel Petr in 1937 as cited in (Schwarz 1956)). Now any solution $v \in B_f$ to (1) must have entries which are a linear combination of the r basis elements of B_f. If v has length t as a vector over B_f, we can treat the coefficients of the combinations as rt unknowns. Then equation (1) can be expanded into a linear system over $\mathbb{F}_q$ in these unknowns, so a basis for the solution space over $\mathbb{F}_q$ can be found in deterministic polynomial time. Specifically, each row of the matrix L contributes n linear relations on the rt unknown coefficients, obtained by considering coefficients of the n basis monomials for R_f over $\mathbb{F}_q$. □

We now suppose that $t(z)$ is an irreducible factor of f. Given $\mathcal{L}$ we denote by $\mathcal{L}_t$ the system of linear equations over R_t obtained by projecting each matrix entry in L under π_t. We shall call this the *projected system* of $\mathcal{L}$ under π_t. Once again we wish to find solutions of $\mathcal{L}_t$ which are vectors over $B_t \cong \mathbb{F}_q$. Certainly any solution v of $\mathcal{L}$ with entries in B_f will be sent under the map on vectors induced by π_t to a solution of $\mathcal{L}_t$ with entries in B_t. Moreover, this solution will be non-zero if and only if $t(z)$ does not divide all of the entries in v thought of as polynomials in $\mathbb{F}_q[z]$. Conversely, any solution to $\mathcal{L}_t$ with entries in B_t can be lifted using the Chinese remainder theorem to a solution for $\mathcal{L}$ with entries in B_f. Precisely, we take the unique lifting of each entry in the solution from B_t to B_f which reduces to zero for all other projections π_{f_j} $(f_j \neq t)$.

Our deterministic factoring method is based upon the following proposition.

PROPOSITION 2.1: *Let $\mathcal{L}$ be any linear system over R_f. Let $S \subseteq \{1, 2, \ldots, r\}$ with the following properties: The dimension over $\mathbb{F}_q$ of the solution space in $B_{f_j} (\cong \mathbb{F}_q)$ of the projected system $\mathcal{L}_{f_j}$ is non-zero if and only if $j \in S$. Then we can compute in deterministic polynomial time the factorisation*

$$f = (\prod_{j \in S} f_j)(\prod_{j \notin S} f_j).$$

Proof: Compute a basis over $\mathbb{F}_q$ for the space of solutions in B_f of the linear system $\mathcal{L}$ over R_f. We claim the greatest common divisor h, say, of f and the polynomials which occur as entries in the basis vectors is exactly

$$\prod_{j \notin S} f_j.$$

To see this, suppose $j \in S$. Then there exists some non-zero solution $\bar{v}$ of the linear system $\mathcal{L}_{f_j}$ which can be lifted to a non-zero solution v of the linear system $\mathcal{L}$, as previous described. This solution of $\mathcal{L}$ must lie in the span of the basis vectors, and thus if f_j divided all the entries in the basis vectors we would have that f_j divides all the entries in v, but then $\bar{v} = 0$ — a contradiction. Hence f_j does not divide the greatest common divisor h. Now suppose that $j \notin S$, and also that f_j does not divide h. Then f_j does not divide all the entries in the basis vectors of the solution space of $\mathcal{L}$. Thus there exists at least one basis element which projects down to a non-zero solution of $\mathcal{L}_{f_j}$ under π_{f_j} — a contradiction. Thus h is as claimed.

Now one may compute the factor h in deterministic polynomial time using only the deterministic algorithm for computing bases for $\mathcal{L}$ from Lemma 2.1, and the euclidean algorithm for greatest common divisors of univariate polynomials. This completes the proof. $\square$

3. Randomised factorization

We present a variation of Kaltofen's algorithm (Kaltofen 1982; von zur Gathen and Kaltofen 1985; Kaltofen 1985). For simplicity, we only give the version for bivariate polynomials, but his algorithm works for polynomials with any number of variables.

Following tradition, we shall give the definition of a *nice* polynomial. We shall say that $f \in \mathbb{F}_q[y, x]$ of total degree n is *nice* if the reduction modulo y of f is squarefree and of degree n. Note that factors of nice polynomials are also nice.

ALGORITHM 3.1 [Randomised Distinct-Degree Factorization]
Input: A nice polynomial $f(y, x) \in \mathbb{F}_q[y, x]$ of total degree n. A positive integer m such that f has no factors of degree strictly less than m.
Output: Nice polynomials $g, h \in \mathbb{F}_q[y, x]$ and integer s which satisfy the following conditions: g is a product of s irreducible polynomials of degree m, and $f = gh$ where h has no factors of degree strictly less than $m + 1$. (Note that s may equal 0.)

Step 0: Set $s \leftarrow 0, g \leftarrow 1, h \leftarrow f$. Factor $f_0(z) = \prod_{i=1}^{r} f_0^{(i)}(z)$ using a randomised algorithm and for each irreducible factor $t(z) := f_0^{(i)}(z)$ do Steps 1 and 2.

Step 1: [Approximate a root of $f(y, x)$ in $R_t[[y]]$ where $R_t = \mathbb{F}_q[z]/(t(z))$.]

Letting $k = (2n-1)n$ and $a_0 := z \bmod t \in R_t$, by Newton iteration compute $a_1, \ldots, a_k \in R_t$ such that

$$f(y, a_0 + a_1 y + \ldots + a_k y^k) \equiv 0 \bmod y^{k+1}.$$

For $0 \le i \le m$ compute

$$\alpha^{(i)} := (a_0 + \ldots + a_k y^k)^i \bmod y^{k+1} \in R_t[y].$$

Step 2: [Try to find a polynomial of degree $\le m$ in $\mathbb{F}_q[y,x]$ for which $\alpha^{(1)}$ is the approximation of one of its roots.]
Compute a basis over $\mathbb{F}_q$ for solutions over $B_t (\cong \mathbb{F}_q)$ of the linear system $\mathcal{L}_t$ over R_t given by

$$\sum_{i=0}^{m} u_i(y)\alpha^{(i)} \equiv 0 \bmod y^{k+1}, \tag{2}$$

where $u_i(y) \in \mathbb{F}_q[y]$, $\deg_y(u_i) \le (m-i)$, and the coefficients of $u_i(y)$ are the unknowns. If there exists a non-zero solution $\{u_i\}$ then it is unique up to scaling by $\mathbb{F}_q$. In which case define

$$u = \sum_{i=0}^{m} u_i(y)x^i,$$

which is necessarily an irreducible factor of f of degree m whose reduction modulo y is divisible by $t(x)$. Now check whether $u|h$, for we may have already found this factor. If so set $g \leftarrow gu, h \leftarrow h/u$ and $s \leftarrow s+1$.

Step 3: Output g, h, s.

The justification of the correctness of this algorithm follows from Theorem 1 in (Kaltofen 1985). Comment must be made however on certain minor aspects which distinguish this algorithm from the version in (Kaltofen 1985). First, we take as an input assumption that f does not have any factors of degree less than m, whereas in (Kaltofen 1985) m is increased from 1 until the first factor is found. Second, we allow solutions to the linear system (2) in which $u_m(y)$ may be zero. In (Kaltofen 1985) the author fixes $u_m(y)$ to be the unity in $\mathbb{F}_q$. We remove this restriction simply to make (2) a homogeneous system so the theory developed in Section 2 directly applies. Indeed any solution will necessarily have $u_m(y) \ne 0$, since factors of nice polynomials are nice. Note that only those irreducible factors $t(z)$ of degree not greater than m can possibly yield non-zero solutions to $\mathcal{L}_t$. Also each such factor can yield at most one non-zero solution, up to scaling, as $t(z)$ can occur as a factor of the reduction modulo y of at most one irreducible factor of f. This justifies the claim on the uniqueness of u.

It is perhaps helpful to also explain precisely how Step 2 of the algorithm relates to Section 2. The linear system $\mathcal{L}_t$ may be made into a more explicit linear system of the form "$Lv = 0$" by equating coefficients of each power of y. In this case the matrix L would be of size $(k+1) \times ((m+1)(m+2)/2)$ with entries from R_t. Note also that the rather unwieldy sentence "Compute a basis ... " could be replaced by "Solve the following linear system over $\mathbb{F}_q$ ... ". However, we choose the more cumbersome version to preserve the analogy with Section 2, and in preparation for Section 4.

As in (Kaltofen 1987) the problem is that the univariate factor $t(z)$ cannot be computed in deterministic polynomial time. Following Kaltofen, our approach is to work in $R_{f_0} = \mathbb{F}_q[z]/(f_0(z))$ and construct an analogous linear system to (2), only with solutions as vectors over the Berlekamp algebra B_{f_0} of $\mathbb{F}_q[z]/(f_0(z))$.

4. Deterministic distinct-degree factorization

We begin with nice polynomials.

Algorithm 4.1 [Deterministic Distinct-Degree Factorization]
Input: A nice polynomial $f(y, x) \in \mathbb{F}_q[y, x]$ of total degree n. A positive integer m such that f has no factors of degree strictly less than m.
Output: Nice polynomials $g, h \in \mathbb{F}_q[y, x]$ and an integer s which satisfy the following conditions: g is a product of s irreducible polynomials of degree m, and $f = gh$ where h has no factors of degree strictly less than $m + 1$. (Note that s may equal 0.)

Step 1: [Approximate a root of $f(y, x)$ in $R_{f_0}[[y]]$ where $R_{f_0} := \mathbb{F}_q[z]/(f_0(z))$]
Define $k := (2n-1)n$ and $a_0 := z \bmod f_0(z) \in R_{f_0}$. By Newton iteration compute $a_1, a_2, \ldots, a_k \in R_{f_0}$ such that

$$f(y, a_0 + a_1 y + \ldots + a_k y^k) \equiv 0 \bmod y^{k+1}.$$

For $0 \le i \le m$ compute

$$\alpha^{(i)} := (a_0 + \ldots + a_k y^k)^i \bmod y^{k+1}.$$

Step 2: [Try to find a polynomial of degree $\le m$ in $B_{f_0}[y, x]$ for which $\alpha^{(1)}$ is the approximation of one of its roots, where B_{f_0} is the Berlekamp subalgebra of R_{f_0}.]
Compute a basis over $\mathbb{F}_q$ of solutions over B_{f_0} to the homogeneous linear system $\mathcal{L}$ over R_{f_0} given by,

$$\sum_{i=0}^{m} u_i(y) \alpha^i \equiv 0 \bmod y^{k+1}, \qquad (3)$$

where the coefficients of $u_i(y) \in B_{f_0}[y]$, $\deg_y(u_i) \le (m-i)$, are the unknowns. If the dimension is zero then output "$s = 0$, $g = 1$, and $h = f$" and halt.

Step 3: Compute the gcd of $f_0(z)$ and the entries of all basis elements of the solution space of $\mathcal{L}$, thought of as polynomials in $\mathbb{F}_q[z]$. Denote this by h_0 and the cofactor of $h_0(z)$ in $f_0(z)$ by $g_0(z)$.

Step 4: We have the factorization $f_0(x) = g_0(x)h_0(x)$. Using Hensel lifting compute a factorization $f = gh$ with $g \equiv g_0 \bmod y$ and $h \equiv h_0 \bmod y$. Output g and h, and also $s := \deg_x(g_0(x))/m$.

Proposition 4.1: *Algorithm 1 outputs correctly and runs in deterministic polynomial time.*

Proof: For each irreducible factor $t(z)$ of $f_0(z)$, the linear system (2) is the projection $\mathcal{L}_t$ of the linear system $\mathcal{L}$ defined by (3) under π_t. (It was briefly explained how to present these linear systems in the form "$Lv = 0$" in the second paragraph following Algorithm 1 and we will not labour this point.) Now $\mathcal{L}_t$ has a non-zero solution if and only if f has a factor of degree not greater than m whose reduction modulo y is divisible by $t(x)$. Such a factor must have degree exactly m and be irreducible by the input assumption on f.

Thus we are in the situation of Proposition 2.1 and we may compute in deterministic polynomial time the factorization

$$f_0(z) = \underbrace{(\prod_{j\in S} f_j)}_{g_0}\underbrace{(\prod_{j\notin S} f_j)}_{h_0}$$

where S is the set of all indices j such that the polynomial $f_j(x)$ is a divisor of the reduction modulo y of an irreducible factor of f of degree m. We have that the first factor $g_0(x)$ is exactly the product of all $f_j(x)$ such that f has an irreducible factor of degree m whose reduction modulo y is divisible by f_j. That is, $g_0(x)$ is just the reduction modulo y of the product g, say, of all irreducible factors of f of degree m. Hence using Hensel lifting we may recover in deterministic polynomial time this factor g and its cofactor h, say, in f. Finally, we can compute the number of irreducible factors of f of degree exactly m as $\deg_x(\prod_{j\in S} f_j(x))/m$. This completes the proof. □

Note that the above algorithm may be used to remove equal-degree irreducible factors of a squarefree univariate polynomial in a somewhat different manner from the usual method (von zur Gathen and Gerhard 1999, Section 14.2). It is illuminating to describe the main features of the algorithm in this special case: Given such a univariate polynomial $f \in \mathbb{F}_q[x] \subseteq \mathbb{F}_q[y, x]$ we have that $f_0 := f \bmod y = f$. In Step 1 of the algorithm the approximate root α of f

in $R_f[[y]]$ is just the exact root $z \bmod f$. In Step 2, in the linear system $\mathcal{L}$ we may ignore higher powers of y, and the problem reduces to finding a sequence of elements $u_0, u_1, \ldots, u_m \in R_f$ such that $\sum_{i=0}^{m} u_i z^i = 0$ in R_f. Now suppose that f_j is an irreducible factor of f of (total) degree m, and write $f_j = \sum_{i=0}^{m} v_i x^i$ where $v_i \in \mathbb{F}_q$. For $0 \leq i \leq m$, define u_i as $(0, 0, \ldots, v_i, \ldots, 0) \in B_f \cong \oplus_{i=1}^{r} \mathbb{F}_q$, where the non-zero entry is in the jth position. Then the sequence u_i gives a solution to our linear system. For this reason essentially, in Step 3 one recovers the product of all irreducible factors of degree m of the univariate polynomial f. Note that the above approach does not seem to lead to an asymptotically faster algorithm than the current best (von zur Gathen and Shoup 1992; Kaltofen and Shoup 1998).

Algorithm 1 may be iterated in a straightforward manner to compute the complete distinct-degree factorisation of a nice bivariate polynomial. That is, one starts with an arbitrary nice f taking $m = 1$ and by repeated application of the algorithm with m incremented by one each time successively remove factors of increasing degree. So we have

Proposition 4.2: *There is an algorithm for computing the distinct-degree factorisation of a nice polynomial in $\mathbb{F}_q[y, x]$ in deterministic polynomial time.*

Now we show how to reduce general polynomials to nice ones. Let $f \in \mathbb{F}_q[y, x]$ of total degree n. By the algorithm of Yun (Yun 1976), one can compute its squarefree decomposition in deterministic polynomial time. Hence we may assume that f is already squarefree in $\mathbb{F}_q[y, x]$.

For small q, say $q < 2n^2$, since Berlekamp's algorithm (Berlekamp 1967; Lidl and Niederreiter 1983) for univariate polynomials runs in deterministic polynomial time, Kaltofen's original version of Algorithm 1 in the previous section can factor f in deterministic polynomial time. So in this case one can certainly find the distinct-degree factorization of f.

Assume that $q \geq 2n^2$. Consider the following substitution:

$$\tilde{f} = f(y + ax + b, x)$$

for some $a, b \in \mathbb{F}_q$. Certainly, any factorization of f gives a factorization of $\tilde{f}$ and vice versa. Note that the coefficient of x^n in $\tilde{f}$ is a polynomial h in a of degree at most n, and

$$\tilde{f}_0 = f(ax + b, x).$$

To make $\tilde{f}$ nice, we just need to pick $a \in \mathbb{F}_q$ such that $h(a) \neq 0$ and then find $b \in \mathbb{F}_q$ such that

$$\mathrm{Resultant}_x(\tilde{f}_0, \frac{\partial}{\partial x}\tilde{f}_0) \neq 0,$$

which is a nonzero polynomial in b of degree at most $n(2n-1) < 2n^2$, since f is squarefree in $\mathbb{F}_q[y, x]$. Hence both a and b can be found after trying at most $2n^2$ elements in $\mathbb{F}_q$. So a and b can be found in deterministic polynomial time. Combining with Proposition 4.2, we have the following result.

Theorem 4.1: *There is an algorithm for computing the distinct-degree factorization of any polynomial in* $\mathbb{F}_q[y, x]$ *in deterministic polynomial time.*

When the distinct-degree factorization of f is computed, it is simple to find the number of irreducible factors of f.

Corollary 4.1: *There is an algorithm for counting the number of irreducible factors of any polynomial in* $\mathbb{F}_q[y, x]$ *in deterministic polynomial time.*

This resolves an open problem posed by Kaltofen in (Kaltofen 1987).

Note that having obtained a distinct total degree factorisation, one may attempt to find finer factorisations by considering different degree orderings. In the algorithm we restrict to the standard degree ordering obtained by giving both variables equal "weight" and defining the (total) degree of a polynomial to be the greatest weight of any monomial. This is an inessential restriction, and our algorithm works with different degree orderings, such as degree in x or degree in y, or other degree orderings in which the two variables are assigned different weights. The easiest way to obtain a nice input while accounting for these degree orders is to work with a new main variable z and factor the tri-variate polynomial $f(az + b + y, z + x)$.

We should remark that the method applies equally well to polynomials with more than two variables where one uses dense input size. The Newton approximation in Step 1 of Algorithm 1 can be carried out similarly as in Kaltofen (Kaltofen 1985) and gcds of multivariate polynomials can be computed in deterministic polynomial time (Brown 1971). Hence one can count the number of irreducible factors of any multivariate polynomial, in the dense representation, in deterministic polynomial time.

References

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

Brown, W. S. (1971). On Euclid's algorithm and the computation of polynomial greatest common divisors. *J. ACM*, 18:478–504.

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

von zur Gathen, J., Gerhard, J. (1999). *Modern Computer Algebra*. Cambridge University Press, Cambridge, New York, Melbourne.

von zur Gathen, J., Kaltofen, E. (1985). Factoring multivariate polynomials over finite fields. *Math. Comput.*, 45:251–261.

von zur Gathen, J., Shoup, V. (1992). Computing Frobenius maps and factoring polynomials. *Comput. Complexity*, 2:187–224.

Kaltofen, E. (1982). A polynomial-time reduction from bivariate to univariate integral polynomial factorization. In *Proc. 23rd Annual Symp. Foundations of Comp. Sci.*, pages 57–64. IEEE. Journal version in Kaltofen (1985).

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

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

Kaltofen, E., Shoup, V. (1998). Subquadratic-time factoring of polynomials over finite fields. *Math. Comput.*, 67(223):1179–1197.

Lidl, R., Niederreiter, H. (1983). *Finite Fields.* Addison-Wesley, Reading, Massachusetts, USA. Now distributed by Cambridge University Press.

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

Yun, D. Y. Y. (1976). On square-free decomposition algorithms. In Jenks, R. D., editor, *Proc. 1976 ACM Symp. Symbolic Algebraic Comput.*, pages 26–35. ACM. SYMSAC was held at IBM Research in Yorktown Heights, New York.