MATHEMATICS OF COMPUTATION
VOLUME 62, NUMBER 206
APRIL 1994, PAGES 819–830

FACTORING POLYNOMIALS OVER FINITE FIELDS USING DIFFERENTIAL EQUATIONS AND NORMAL BASES

HARALD NIEDERREITER

ABSTRACT. The deterministic factorization algorithm for polynomials over finite fields that was recently introduced by the author is based on a new type of linearization of the factorization problem. The main ingredients are differential equations in rational function fields and normal bases of field extensions. For finite fields of characteristic 2, it is known that this algorithm has several advantages over the classical Berlekamp algorithm. We develop the algorithm in a general framework, and we show that it is feasible for arbitrary finite fields, in the sense that the linearization can be achieved in polynomial time.

1. INTRODUCTION

Like the classical algorithm of Berlekamp [1], the recently developed deterministic algorithm of the author [12] for factoring polynomials over finite fields is based on a linearization of the factorization problem, i.e., on a reduction to a system of linear equations. However, the new algorithm uses a completely different procedure to achieve the linearization, a key step being the consideration of differential equations in rational function fields. The algorithm introduced in [12] was restricted to squarefree polynomials over finite prime fields, but it was shown by Miller [8] that the condition that the polynomial be squarefree can be dropped.

In [13] the author has extended the factorization algorithm to arbitrary finite fields. Again, the algorithm readily applies to arbitrary polynomials; i.e., no prior reduction to the squarefree case is necessary. The analysis in [13] revealed that, at least for finite fields of characteristic 2, the new algorithm has several advantages over the Berlekamp algorithm. In [13] one can, in fact, find two ways of generalizing the algorithm in [12] to arbitrary finite fields: one method uses normal bases of field extensions, and the other Hasse-Teichmüller derivatives. The latter approach was further pursued by Niederreiter and Göttfert [14], who showed that it leads to a feasible linearization technique for arbitrary finite fields. Thus, the new algorithm looks more and more like a method that is not only of interest for small finite fields, as suggested initially in [12]. This will be borne out again by the results of the present paper which, in particular, demonstrate the usefulness of the algorithm for large finite fields of small characteristic.

Received by the editor November 13, 1992.
1991 *Mathematics Subject Classification.* Primary 11T06, 11Y16.
Key words and phrases. Polynomial factorization, differential equations in rational function fields, normal bases.

0025-5718/94 $1.00 + $.25 per page

In this paper it is demonstrated that the method in [13] using normal bases yields a feasible linearization, in the sense that the coefficient matrix of the appropriate system of linear equations can be calculated in polynomial time. We also combine the approach employing normal bases with the method using Hasse-Teichmüller derivatives to obtain a generalization of Algorithm B in [13]. In §2 we describe the general scheme of the algorithm; some of the underlying principles are valid for any field of positive characteristic. The linearization of the factorization problem proceeds in two steps: first from a suitable differential equation to a system of algebraic equations, and then from this system of algebraic equations to a system of linear equations. In §3 we analyze the first step and show how to efficiently compute the coefficient matrix of the linear part in the system of algebraic equations. For this purpose, the basic technical tool in [14], namely the analysis of a certain decimation operator on sequences, is exploited further. In §4 we analyze the transition from a system of algebraic equations to a system of linear equations by using normal bases of extensions of finite fields. The cost of setting up the coefficient matrix of the system of linear equations is bounded, in particular, in terms of the complexity of the normal basis. The bound shows that low-complexity normal bases are preferable in this context.

2. Description of the algorithm

We first need to define Hasse-Teichmüller derivatives of formal Laurent series (see [5, 15]). For an arbitrary field F let $F((x^{-1}))$ be the field of formal Laurent series over F in the variable x^{-1}. The elements of $F((x^{-1}))$ have the form

$$\sum_{n=w}^{\infty} s_n x^{-n},$$

where w is an arbitrary integer and all $s_n \in F$. For an integer $k \geq 0$ the *Hasse-Teichmüller derivative* $H^{(k)}$ of order k is defined on $F((x^{-1}))$ by

$$H^{(k)}\left(\sum_{n=w}^{\infty} s_n x^{-n}\right) = \sum_{n=w}^{\infty} \binom{-n}{k} s_n x^{-n-k}.$$

Since $F((x^{-1}))$ contains the rational function field $F(x)$ as a subfield, $H^{(k)}$ is thus in particular defined on $F(x)$. We note that $H^{(k)}$ is an F-linear operator on $F((x^{-1}))$.

The starting point of the factorization algorithm is the consideration of a certain differential equation in terms of Hasse-Teichmüller derivatives. Let F be a field of prime characteristic p, and let $r > 1$ be a power of p. Then the differential equation in question is

$$H^{(r-1)}(y) = y^r \tag{1}$$

with an unknown $y \in F((x^{-1}))$. For the application to factorization we are, in fact, interested only in rational solutions $y \in F(x)$ of (1). The differential equation (1) was introduced and studied in [13]. In the case $r = p$ we have $H^{(p-1)}(y) = -y^{(p-1)}$ for all $y \in F((x^{-1}))$, where $y^{(p-1)}$ is the ordinary derivative of y of order $p-1$, and so (1) reduces to a differential equation considered earlier in [12].

Let $r > 1$ again be a power of the prime p. We denote by F_r the finite field of order r. If the field F contains F_r as a subfield, then $J(y) = H^{(r-1)}(y) - y^r$ is an F_r-linear operator on $F((x^{-1}))$, and therefore the solutions of (1) form an F_r-linear subspace of $F((x^{-1}))$. The rational solutions of (1) can be described explicitly by the following result, which is a special case of [13, Theorem 2].

Lemma 1. *Suppose the perfect field F contains the finite field F_r as a subfield, and let $f \in F[x]$ be a monic nonconstant polynomial. Then the solutions y of (1) of the form $y = h/f$ with $h \in F[x]$ are exactly given by*

$$y = \sum_{i=1}^{m} c_i \frac{g_i'}{g_i} \quad \text{with } c_1, \ldots, c_m \in F_r,$$

where $g_1, \ldots, g_m \in F[x]$ are the distinct monic irreducible factors in the canonical factorization of f over F.

This result is applied to factorization in the following way. Let F and $f \in F[x]$ be as in Lemma 1, where f is the polynomial to be factored. Suppose we have a computationally feasible method for finding the solutions y of (1) of the form $y = h/f$ with $h \in F[x]$. Note that according to Lemma 1, the differential equation (1) has exactly r^m such solutions. It follows from the form of the solutions in Lemma 1 that if $y = h/f$ solves (1), then

$$\frac{f}{\gcd(f, h)} = \prod_{\substack{i=1 \\ c_i \neq 0}}^{m} g_i. \tag{2}$$

Thus, if h runs through all r^m numerator polynomials of the solutions $y = h/f$ of (1), then $f/\gcd(f, h)$ yields all 2^m monic factors of the squarefree part $g_1 \cdots g_m$ of f (with repetitions if $r > 2$). In particular, we get in this way all monic irreducible factors $g_1, \ldots, g_m$ of f, which readily yields the complete canonical factorization of f. In an alternative strategy, we just strive to get one nontrivial factor of f out of (2), and then we apply the factorization algorithm again to this nontrivial factor and its complementary factor of f and iterate. Compare also with the discussion in [13].

The potential bottleneck in the above procedure for determining $g_1, \ldots, g_m$ is the calculation of the r^m gcd's in (2). For random polynomials f over finite fields and r and $d = \deg(f)$ of reasonable size, this problem is not so serious since the average order of magnitude of the number m of distinct monic irreducible factors of f is small, namely $\log d$ according to [7, §6.2.4]. However, in unfavorable situations, m can be close to d, and then difficulties arise already for moderately large d. We remark that in the frequently encountered case $r = 2$, Göttfert [4] has recently shown that, by a more refined approach, the number of required gcd calculations for the determination of $g_1, \ldots, g_m$ can be reduced from r^m to $O(m^2)$. This leads to a polynomial-time factorization algorithm over finite fields of characteristic 2.

The central problem that remains to be discussed is how to actually solve the differential equation (1). Let F be an arbitrary field of prime characteristic p, let $r > 1$ be a power of p, and let $f \in F[x]$ be monic with $\deg(f) = d \geq 1$. We are interested in the solutions of (1) of the form $y = h/f$ with f fixed and

$h \in F[x]$ unknown, and so we write (1) as

$$f^r H^{(r-1)}\left(\frac{h}{f}\right) = h^r. \tag{3}$$

It was shown in [13] that any solution h of (3) satisfies $\deg(h) < d$, so that we can write $h(x) = \sum_{k=0}^{d-1} h_k x^k$ with all $h_k \in F$. Also, by the arguments following [13, Eq. (19)] we know that both sides of (3) are polynomials over F of degrees $\leq (d-1)r$ and that both sides of (3) are polynomials in x^r. Thus, (3) holds if and only if the coefficients of x^{jr}, $0 \leq j \leq d-1$, agree on both sides. The comparison of coefficients yields a system of d algebraic equations for the unknowns $h_0, \ldots, h_{d-1}$. If $N_r(f)$ denotes the $d \times d$ coefficient matrix over F on the (linear) left-hand side of (3), then (3) is equivalent to the system

$$N_r(f)\mathbf{h}^{\mathsf{T}} = (\mathbf{h}^r)^{\mathsf{T}}, \tag{4}$$

where $\mathbf{h} = (h_0, \ldots, h_{d-1}) \in F^d$ is the coefficient vector of h, and $\mathbf{h}^r$ stands for the vector $(h_0^r, \ldots, h_{d-1}^r) \in F^d$. An efficient method for calculating the matrix $N_r(f)$ will be developed in §3. We will then have achieved a computationally feasible transition from (3) to (4).

The next step is to linearize (4). For this purpose, we now assume that F is an arbitrary *finite* field, say $F = F_q$. Let F_r be a subfield of F_q; then all the theory described above applies. In most practical implementations one will take $r = p$, the characteristic of F_q, so that F_r is the prime subfield of F_q, but the case of an arbitrary finite extension F_q/F_r of finite fields that is treated here is also of interest. A convenient linearization of (4) is achieved by utilizing normal bases of the extension. This method was already developed in [13] for the special case $r = p$, and it can be generalized in an obvious fashion. If $q = r$, then $\mathbf{h}^r = \mathbf{h}$, and the system (4) is already linear. Thus, we can assume that $q = r^t$ with an integer $t \geq 2$.

We recall that a *normal basis* of F_q over F_r is an ordered basis of F_q over F_r of the form $\{\alpha, \alpha^r, \alpha^{r^2}, \ldots, \alpha^{r^{t-1}}\}$ with some $\alpha \in F_q$. A normal basis exists for any extension F_q/F_r (see [6, Theorem 2.35]). Now we fix a normal basis B of F_q over F_r and write the entries of the matrix $N_r(f)$ and the unknowns h_k, $0 \leq k \leq d-1$, as linear combinations of the elements of B with coefficients from F_r. For $0 \leq k \leq d-1$ let $h_k^{(i)} \in F_r$, $0 \leq i \leq t-1$, be the coefficients in the representation of h_k. We insert all these linear combinations into (4), and on the left-hand side we express each product of elements of B as a linear combination of the elements of B. Finally, we carry out a comparison of coefficients of the elements of B on both sides of the resulting equations. In this way we arrive at the system of homogeneous linear equations

$$K_{q,r}(f, B)\mathbf{H}^{\mathsf{T}} = \mathbf{0} \in F_r^{dt}, \tag{5}$$

where $K_{q,r}(f, B)$ is a $dt \times dt$ matrix over F_r and $\mathbf{H} \in F_r^{dt}$ contains the unknowns $h_k^{(i)}$ in some order. For the sake of definiteness, we put the $h_k^{(i)}$ in lexicographic order, that is, $h_0^{(0)}, \ldots, h_0^{(t-1)}, h_1^{(0)}, \ldots, h_1^{(t-1)}, \ldots$. The system (5) is equivalent to the system (4), and hence equivalent to the differential equation (3). This provides the desired linearization. In principle, this linearization works also if we use an arbitrary ordered basis of F_q over F_r, but

the choice of a normal basis simplifies the treatment of the right-hand side of (4) considerably.

The facts that (3) and (5) are equivalent and that (3) has exactly r^m solutions imply that the matrix $K_{q,r}(f, B)$ in (5) has rank $dt-m$. In §4 we will analyze the cost of deriving the coefficient matrix $K_{q,r}(f, B)$ of (5) from the matrix $N_r(f)$ and also the total setup cost of $K_{q,r}(f, B)$. We note that if $r = p$, then $K_{q,r}(f, B)$ is the same as the matrix $K_q(f, B)$ introduced in [13].

3. From the differential equation to algebraic equations

In this section we analyze the transition from the differential equation (3) to the system (4) of algebraic equations. The main result of this section (Theorem 2) shows that the matrix $N_r(f)$ can be calculated in polynomial time, where the time unit is taken to be one arithmetic operation in the underlying field.

We consider (3) under the conditions stipulated in §2, namely that F is an arbitrary field of prime characteristic p, that $r > 1$ is a power of p, and that $f \in F[x]$ is monic with $\deg(f) = d \geq 1$. Let F^∞ be the vector space (over F) of all sequences of elements of F under termwise operations. We abbreviate a sequence $s_0, s_1, \ldots$ of elements of F by (s_n). We define two linear operators on F^∞, the *shift operator* $T(s_n) = (s_{n+1})$ and the *decimation operator* $\Delta_r(s_n) = (s_{nr})$. Let $S(f)$ be the kernel of the linear operator $f(T)$ on F^∞. The restriction of Δ_r to $S(f)$ is again denoted by Δ_r. If $f(x) = \sum_{j=0}^{d} a_j x^j$ with all $a_j \in F$, then we put

$$f_r(x) = \sum_{j=0}^{d} a_j^r x^j \in F[x].$$

The following basic lemma is a special case of [11, Corollary 1], but we include its simple proof for the sake of completeness.

Lemma 2. *The operator Δ_r is a linear transformation from $S(f)$ into $S(f_r)$.*

Proof. It suffices to show that $\Delta_r\sigma \in S(f_r)$ for all $\sigma \in S(f)$. Since $f(T)$ is the zero operator on $S(f)$, and $f(x)$ divides $f(x)^r = f_r(x^r)$, it follows that $f_r(T^r)$ is the zero operator on $S(f)$. Thus, for any $\sigma \in S(f)$ we get

$$f_r(T)\Delta_r\sigma = \Delta_r f_r(T^r)\sigma = 0 \in F^\infty,$$

which means that $\Delta_r\sigma \in S(f_r)$. □

Every sequence from $S(f)$ satisfies a d-term linear recurrence relation with characteristic polynomial f. Consequently, each element (s_n) of $S(f)$ is uniquely determined by its *initial state vector* $(s_0, \ldots, s_{d-1}) \in F^d$, and we have $\dim(S(f)) = d$. If with each $\sigma = (s_n) \in F^\infty$ we associate its *generating function*

$$\sum_{n=0}^{\infty} s_n x^{-n-1} \in F((x^{-1})),$$

then by [10, Lemma 1], or by a straightforward verification, we obtain that $\sigma \in S(f)$ if and only if the generating function y of σ has the form $y = h/f$ with $h \in F[x]$ and $\deg(h) < d$.

For an ordered basis E of $S(f)$, let the row vector $[\sigma]_E \in F^d$ be the coordinate vector of $\sigma \in S(f)$ relative to E. For another monic polynomial $g \in F[x]$ with $\deg(g) = d$, let E' be an ordered basis of $S(g)$. Then for a linear transformation A from $S(f)$ into $S(g)$, we denote by $[A]_{E,E'}$ the $d \times d$ matrix over F representing A relative to E and E', in the sense that

$$[A\sigma]_{E'}^{\mathrm{T}} = [A]_{E,E'}[\sigma]_E^{\mathrm{T}} \quad \text{for all } \sigma \in S(f).$$

As in [14], we work with two special ordered bases of $S(f)$. Let $\omega = (u_n)$ be the *impulse response sequence* with characteristic polynomial f, i.e., the sequence from $S(f)$ with initial state vector $(0, \ldots, 0, 1)$. If $\omega_k = T^k\omega$ for $0 \le k \le d-1$, then

$$E_1 = E_1(f) = \{\omega_0, \ldots, \omega_{d-1}\}$$

is an ordered basis of $S(f)$, called the *canonical basis.* The generating function of ω_k is $x^k/f(x)$. As noted above, any $\sigma \in S(f)$ has a generating function of the form $h(x)/f(x)$ with $h(x) = \sum_{k=0}^{d-1} h_k x^k$ and all $h_k \in F$. Then

$$\frac{h(x)}{f(x)} = \sum_{k=0}^{d-1} h_k \frac{x^k}{f(x)},$$

and so, by turning to the corresponding sequences, we obtain

$$\sigma = \sum_{k=0}^{d-1} h_k \omega_k.$$

In other words, we have

$$[\sigma]_{E_1} = (h_0, \ldots, h_{d-1}), \tag{6}$$

so that the coordinate vector of σ relative to E_1 is the coefficient vector of the numerator polynomial of the generating function of σ.

The second special ordered basis of $S(f)$ is the *standard basis*

$$E_2 = E_2(f) = \{\varepsilon_0, \ldots, \varepsilon_{d-1}\}.$$

Here the ε_k, $0 \le k \le d-1$, are the sequences from $S(f)$ whose initial state vectors are the standard basis vectors of F^d in their natural order. Thus, ε_0 has the initial state vector $(1, 0, \ldots, 0)$, ε_1 has the initial state vector $(0, 1, 0, \ldots, 0)$, and so on. If $\sigma = (s_n)$ is an arbitrary sequence from $S(f)$, then

$$\sigma = \sum_{k=0}^{d-1} s_k \varepsilon_k.$$

Thus,

$$[\sigma]_{E_2} = (s_0, \ldots, s_{d-1}), \tag{7}$$

so that the coordinate vector of σ relative to E_2 is the initial state vector of σ. We can now give the following formula for the matrix $N_r(f)$ in (4).

Lemma 3. *We have* $N_r(f) = [\Delta_r]_{E_1, E_1'}$ *with* $E_1 = E_1(f)$ *and* $E_1' = E_1(f_r)$.

Proof. If $N_r(f) = (n_{jk})_{0 \le j,k \le d-1}$ and $E_1 = \{\omega_0, \ldots, \omega_{d-1}\}$, then we have to show that

$$[\Delta_r \omega_k]_{E_1'} = (n_{0k}, n_{1k}, \ldots, n_{d-1,k}) \quad \text{for } 0 \le k \le d-1. \tag{8}$$

Writing $h(x) = \sum_{k=0}^{d-1} h_k x^k$ with all $h_k \in F$ on the left-hand side of (3), we obtain

$$f^r H^{(r-1)}\left(\frac{h}{f}\right) = \sum_{k=0}^{d-1} f(x)^r H^{(r-1)}\left(\frac{x^k}{f(x)}\right) h_k.$$

Thus, it follows from the definition of $N_r(f)$ that n_{jk} is the coefficient of x^{jr} in $f(x)^r H^{(r-1)}(x^k/f(x))$. By the discussion after (3), $f(x)^r H^{(r-1)}(x^k/f(x))$ is of degree $\le (d-1)r$ and a polynomial in x^r. Therefore,

$$f(x)^r H^{(r-1)}\left(\frac{x^k}{f(x)}\right) = \sum_{j=0}^{d-1} n_{jk} x^{jr} \quad \text{for } 0 \le k \le d-1. \tag{9}$$

From the proof of [14, Theorem 2.1] it follows that if y is the generating function of an arbitrary $(s_n) \in F^\infty$, and if z is the generating function of $\Delta_r(s_n) = (s_{nr})$, then

$$H^{(r-1)}(y) = \sum_{n=0}^{\infty} s_{nr} x^{-nr-r} = \sum_{n=0}^{\infty} s_{nr} (x^r)^{-n-1} = z(x^r). \tag{10}$$

If for $0 \le k \le d-1$ we let $y_k = x^k/f(x)$ be the generating function of ω_k, and z_k the generating function of $\Delta_r \omega_k$, then by (10) and (9) we obtain

$$\begin{aligned} z_k(x^r) = H^{(r-1)}(y_k) = H^{(r-1)}\left(\frac{x^k}{f(x)}\right) &= \frac{1}{f(x)^r} \sum_{j=0}^{d-1} n_{jk} x^{jr} \\ &= \frac{1}{f_r(x^r)} \sum_{j=0}^{d-1} n_{jk} x^{jr}, \end{aligned}$$

and so

$$z_k = \frac{1}{f_r(x)} \sum_{j=0}^{d-1} n_{jk} x^j \quad \text{for } 0 \le k \le d-1.$$

In view of (6) this shows (8). □

A formula for $N_r(f)$ that is more explicit than that in Lemma 3 can now be derived in Theorem 1 below. We define three more $d \times d$ matrices over F as follows. The matrix $G(f_r)$ is obtained from the coefficients of the polynomial f_r, namely

$$G(f_r) = \begin{pmatrix} a_1^r & a_2^r & a_3^r & \cdots & a_{d-1}^r & a_d^r \\ a_2^r & a_3^r & a_4^r & \cdots & a_d^r & 0 \\ a_3^r & a_4^r & a_5^r & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_d^r & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

The Berlekamp matrix $B_r(f) = (b_{jk})_{0\le j,k\le d-1}$ is determined by the congruences

$$x^{jr} \equiv \sum_{k=0}^{d-1} b_{jk} x^k \pmod{f(x)} \quad \text{for } 0 \le j \le d-1. \tag{11}$$

Finally, we introduce the Hankel matrix

$$U(f) = (u_{j+k})_{0\le j,k\le d-1},$$

where $(u_n) = \omega$ is the impulse response sequence with characteristic polynomial f.

Theorem 1. *We have* $N_r(f) = G(f_r)B_r(f)U(f)$.

Proof. Let I be the identity operator on $S(f)$, let E_1 be the canonical basis and E_2 the standard basis of $S(f)$, and let I', E_1', and E_2' be the corresponding objects for $S(f_r)$. Then we have the linear algebra identity

$$[\Delta_r]_{E_1,E_1'} = [I']_{E_2',E_1'}[\Delta_r]_{E_2,E_2'}[I]_{E_1,E_2}. \tag{12}$$

If $E_2' = \{\delta_0, \dots, \delta_{d-1}\}$, then for $0 \le k \le d-1$ the generating function v_k of δ_k has the form

$$v_k = x^{-k-1} + \sum_{n=d}^{\infty} s_n x^{-n-1}$$

with $s_n \in F$ for all $n \ge d$. Since $f_r v_k$ must be a polynomial of degree $< d$, we have

$$f_r v_k = \left(\sum_{j=0}^{d} a_j^r x^j\right)\left(x^{-k-1} + \sum_{n=d}^{\infty} s_n x^{-n-1}\right) = \sum_{j=0}^{d-k-1} a_{j+k+1}^r x^j,$$

and so

$$v_k = \frac{1}{f_r(x)} \sum_{j=0}^{d-k-1} a_{j+k+1}^r x^j.$$

On account of (6) this yields

$$[\delta_k]_{E_1'} = (a_{k+1}^r, a_{k+2}^r, \dots, a_d^r, 0, \dots, 0) \quad \text{for } 0 \le k \le d-1.$$

This shows that

$$[I']_{E_2',E_1'} = G(f_r). \tag{13}$$

Now we fix an element ε_k, $0 \le k \le d-1$, of the ordered basis E_2 and let $\varepsilon_k = (t_n)$. Since (11) implies that $x^{jr} - \sum_{l=0}^{d-1} b_{jl}x^l$ is a characteristic polynomial of ε_k, we have

$$t_{n+jr} = \sum_{l=0}^{d-1} b_{jl} t_{n+l} \quad \text{for } n \ge 0 \text{ and } 0 \le j \le d-1.$$

In particular, we obtain

$$t_{jr} = \sum_{l=0}^{d-1} b_{jl} t_l = b_{jk} \quad \text{for } 0 \le j \le d-1,$$

where we used the special form of the initial state vector of ε_k in the second identity. In view of (7) this shows that

$$[\Delta_r \varepsilon_k]_{E_2'} = (b_{0k}, b_{1k}, \ldots, b_{d-1,k}) \quad \text{for } 0 \le k \le d-1,$$

and so we obtain

$$[\Delta_r]_{E_2, E_2'} = B_r(f). \tag{14}$$

It is clear that $[I]_{E_1, E_2} = U(f)$. Together with (12), (13), (14), and Lemma 3 this yields the desired result. □

Theorem 2. *The setup cost for the matrix $N_r(f)$ is $O(d^e + (d^2 + d\log r)L(d))$ arithmetic operations in F, where $d = \deg(f)$, $L(d) = (\log d)\log\log d$, and $e < 2.4$ is the exponent of fast matrix multiplication.*

Proof. It requires $O(d + \log r)$ polynomial multiplications mod f to set up the matrix $B_r(f)$. By a result of Cantor and Kaltofen [3], each polynomial multiplication mod f can be performed using $O(dL(d))$ arithmetic operations in F. Thus, the setup cost for $B_r(f)$ is $O((d^2 + d\log r)L(d))$ arithmetic operations in F. Each coefficient of the polynomial f_r can be calculated by $O(\log r)$ arithmetic operations in F, and so the setup cost for the matrix $G(f_r)$ is $O(d\log r)$ arithmetic operations in F. In the matrix $U(f)$, only the terms $u_0, u_1, \ldots, u_{2d-2}$ of the sequence ω appear as entries. Since ω satisfies a d-term linear recurrence relation, the setup cost for $U(f)$ is $O(d^2)$ arithmetic operations in F. The desired result follows now from Theorem 1. □

We remark that in the important special case $r = 2$ there is no setup cost for $N_r(f)$, as was already shown in [12] (note that $N_2(f)$ has the same form as the matrix $M_2(f)$ in [12]).

4. From algebraic equations to linear equations

The procedure of reducing the system (4) of algebraic equations to the system (5) of linear equations by using normal bases was already described in §2. Here we analyze the cost of setting up the coefficient matrix $K_{q,r}(f, B)$ of (5).

As in §2, we consider a finite extension F_q/F_r of finite fields, where $q = r^t$ with $t \ge 2$, and we let $f \in F_q[x]$ be monic with $\deg(f) = d \ge 1$. Let $B = \{\alpha, \alpha^r, \alpha^{r^2}, \ldots, \alpha^{r^{t-1}}\}$ with $\alpha \in F_q$ be a fixed normal basis of F_q over F_r. Let the multiplication table of the basis B be given by

$$\alpha^{r^i}\alpha^{r^l} = \sum_{b=0}^{t-1} c(i, l, b)\alpha^{r^b} \quad \text{for } 0 \le i, l \le t-1, \tag{15}$$

where all $c(i, l, b) \in F_r$. The *complexity* $C(B)$ of B is defined as the number of ordered pairs (l, b) with $0 \le l, b \le t-1$ for which $c(0, l, b) \ne 0$. This concept was introduced in [9]; see also [2, Chapters 4, 5]. It follows easily that the number of ordered triples (i, l, b) with $0 \le i, l, b \le t-1$ for which $c(i, l, b) \ne 0$ is given by $C(B)t$ (compare with [2, §5.1]). It is trivial that $C(B) \le t^2$. On the other hand, we always have $C(B) \ge 2t-1$, and equality is possible in some cases (see [2, Chapter 5; 9]).

Theorem 3. *Given the matrix $N_r(f)$ over F_q, the setup cost for the matrix $K_{q,r}(f,B)$ is $O(d^2C(B)t)$ arithmetic operations in F_r, where $d=\deg(f)$, $C(B)$ is the complexity of the normal basis B of F_q over F_r, and t is the degree of the extension F_q/F_r.*

Proof. With $N_r(f)=(n_{jk})_{0\le j,k\le d-1}$ the system (4) can be written in the form

$$\sum_{k=0}^{d-1} n_{jk}h_k = h_j^r \quad \text{for } 0 \le j \le d-1. \tag{16}$$

If the coefficients $n_{jk} \in F_q$ are not yet given by their coordinate vectors relative to B, but rather by their coordinate vectors relative to another basis of F_q over F_r, then the transition from the latter basis to the normal basis B is effected by multiplication with a $t \times t$ basis change matrix over F_r. The coordinate vectors (relative to B) of all n_{jk} can thus be computed by using $O(d^2t^2)$ arithmetic operations in F_r. Since $C(B) \ge 2t-1$, this cost can be incorporated in the complexity bound of the theorem. Thus, we can assume that the n_{jk} are available in their basis representations

$$n_{jk} = \sum_{i=0}^{t-1} n_{jk}^{(i)}\alpha^{r^i} \quad \text{for } 0 \le j, k \le d-1,$$

where all $n_{jk}^{(i)} \in F_r$. We can also write the unknowns h_k in the form

$$h_k = \sum_{i=0}^{t-1} h_k^{(i)}\alpha^{r^i} \quad \text{for } 0 \le k \le d-1,$$

where all $h_k^{(i)} \in F_r$. Now we insert these expressions into (16). On the right-hand side, note that the coordinate vector of h_j^r relative to B is obtained from the coordinate vector of h_j relative to B by a cyclic shift to the right, and so no calculations are needed on that side. On the left-hand side, we get for fixed $0 \le j, k \le d-1$, by using (15),

$$\begin{aligned} n_{jk}h_k &= \left(\sum_{i=0}^{t-1} n_{jk}^{(i)}\alpha^{r^i}\right)\left(\sum_{i=0}^{t-1} h_k^{(i)}\alpha^{r^i}\right) = \sum_{i,l=0}^{t-1} n_{jk}^{(i)}h_k^{(l)} \sum_{b=0}^{t-1} c(i,l,b)\alpha^{r^b} \\ &= \sum_{b=0}^{t-1}\left(\sum_{i,l=0}^{t-1} n_{jk}^{(i)}c(i,l,b)h_k^{(l)}\right)\alpha^{r^b}. \end{aligned}$$

Recall that the matrix $K_{q,r}(f,B)$ is obtained by carrying out a comparison of the coefficients of the basis elements α^{r^b}, $0 \le b \le t-1$, on both sides of (16). Thus, a typical coefficient of an unknown $h_k^{(l)}$ on the left-hand side is

$$\sum_{i=0}^{t-1} n_{jk}^{(i)}c(i,l,b).$$

These expressions have to be calculated for all choices of $0 \le j, k \le d-1$

and for all choices of $0 \leq l, b \leq t-1$. If we fix again j and k and vary l and b, then we need $O(C(B)t)$ arithmetic operations in F_r to calculate the corresponding expressions. Finally, if we vary j and k also, then we see that the setup cost for $K_{q,r}(f, B)$ is $O(d^2C(B)t)$ arithmetic operations in F_r. □

Theorem 3 shows that it is advantageous to work with a low-complexity normal basis B. Since we always have $C(B) \leq t^2$, it follows that even with a bad choice of B the cost of deriving $K_{q,r}(f, B)$ from $N_r(f)$ is only $O(d^2t^3)$ arithmetic operations in F_r. If $r = 2$ and $f \in F_q[x]$ has at most s nonzero coefficients, then by the special form of $N_2(f)$ (compare with [12, 13]) the complexity bound in Theorem 3 reduces to $O(sC(B)t + dt)$ arithmetic operations in F_2. This is also the total setup cost for $K_{q,2}(f, B)$ since there is no cost for setting up $N_2(f)$. In general, the total setup cost for $K_{q,r}(f, B)$ is given as follows.

Theorem 4. *The total setup cost for the matrix* $K_{q,r}(f, B)$ *is*

$$O(d^e + (d^2 + d\log r)L(d))$$

arithmetic operations in F_q *and* $O(d^2C(B)t)$ *arithmetic operations in* F_r, *where* $d = \deg(f)$, $L(d)$ *and* e *are as in Theorem* 2, *and* $C(B)$ *and* t *are as in Theorem* 3.

Proof. Apply Theorem 2 with $F = F_q$ and combine it with Theorem3. □

Bibliography

1. E. R. Berlekamp, *Factoring polynomials over finite fields*, Bell System Tech. J. **46** (1967), 1853–1859.
2. I. Blake, X. H. Gao, A. Menezes, R. Mullin, S. Vanstone, and T. Yaghoobian, *Applications of finite fields*, Kluwer Acad. Publ., Boston, 1993.
3. D. G. Cantor and E. Kaltofen, *On fast multiplication of polynomials over arbitrary algebras*, Acta Inform. **28** (1991), 693–701.
4. R. Göttfert, *An acceleration of the Niederreiter factorization algorithm in characteristic* 2, Math. Comp. **62** (1994), 831–839.
5. H. Hasse, *Theorie der höheren Differentiale in einem algebraischen Funktionenkörper mit vollkommenem Konstantenkörper bei beliebiger Charakteristik*, J. Reine Angew. Math. **175** (1936), 50–54.
6. R. Lidl and H. Niederreiter, *Finite fields*, Addison-Wesley, Reading, MA, 1983.
7. M. Mignotte, *Mathematics for computer algebra*, Springer-Verlag, New York, 1992.
8. V. S. Miller, *On the factorization method of Niederreiter*, preprint, 1992.
9. R. C. Mullin, I. M. Onyszchuk, S. A. Vanstone, and R. M. Wilson, *Optimal normal bases in* $GF(p^n)$, Discrete Appl. Math. **22** (1988/89), 149–161.
10. H. Niederreiter, *Sequences with almost perfect linear complexity profile*, Advances in Cryptology—EUROCRYPT '87 (D. Chaum and W. L. Price, eds.), Lecture Notes in Comput. Sci., vol. 304, Springer-Verlag, Berlin, 1988, pp. 37–51.
11. ———, *A simple and general approach to the decimation of feedback shift-register sequences*, Problems Control Inform. Theory/Problemy Upravlen. Teor. Inform. **17** (1988), 327–331.
12. ———, *A new efficient factorization algorithm for polynomials over small finite fields*, Applicable Algebra in Engrg. Comm. Comp. **4** (1993), 81–87.

13. ———, *Factorization of polynomials and some linear-algebra problems over finite fields*, Linear Algebra Appl. **192** (1993), 301–328.

14. H. Niederreiter and R. Göttfert, *Factorization of polynomials over finite fields and characteristic sequences*, J. Symbolic Comput. (to appear).

15. O. Teichmüller, *Differentialrechnung bei Charakteristik p*, J. Reine Angew. Math. **175** (1936), 89–99.

INSTITUTE FOR INFORMATION PROCESSING, AUSTRIAN ACADEMY OF SCIENCES, SONNENFELSGASSE 19, A-1010 VIENNA, AUSTRIA

E-mail address: nied@qiinfo.oeaw.ac.at