MATHEMATICS OF COMPUTATION
VOLUME 62, NUMBER 206
APRIL 1994, PAGES 831–839

AN ACCELERATION OF THE NIEDERREITER FACTORIZATION ALGORITHM IN CHARACTERISTIC 2

RAINER GÖTTFERT

ABSTRACT. A new deterministic factorization algorithm for polynomials over finite fields was recently developed by Niederreiter. The bottleneck in this algorithm is the last stage in which the irreducible factors of the polynomial are derived from the solutions of a system of linear equations. In this paper, we consider finite fields of characteristic 2, and we show that in this case there is a more efficient approach to the last stage of the Niederreiter algorithm, which speeds up the algorithm considerably.

1. INTRODUCTION AND BACKGROUND

A new factorization algorithm for polynomials over finite prime fields was recently developed by Niederreiter [9] and soon generalized to fields of prime characteristic [10, 11]. For further work on this algorithm we refer to Fleischmann [3], Lee and Vanstone [5], Miller [8], and Niederreiter and Göttfert [12]. This paper deals primarily with the important special case of the *Niederreiter algorithm* in which the underlying field is a finite field of characteristic 2. But for the present, let F be an arbitrary perfect field of characteristic 2.

Let $f \in F[x]$ be a monic polynomial of degree $\deg(f) = d \geq 1$ with its canonical factorization

$$f = g_1^{e_1} \cdots g_m^{e_m} \tag{1}$$

over F, i.e., $g_1, \ldots, g_m$ are distinct monic irreducible polynomials in $F[x]$ and $e_1, \ldots, e_m$ are positive integers. The polynomial f is given as

$$f(x) = f_d x^d + f_{d-1} x^{d-1} + \cdots + f_1 x + f_0 \in F[x] \tag{2}$$

with $f_d = 1$. To factor f means to determine the representation (1) from the representation (2).

The core of the characteristic 2 version of the Niederreiter algorithm is the differential equation

$$(fh)' = h^2, \tag{3}$$

where $f \in F[x]$ is the given polynomial to be factored and $h \in F[x]$ an unknown polynomial to be determined. If h_1 and h_2 are polynomials in $F[x]$ that satisfy (3), then so does $h_1 + h_2$. Therefore, the solutions h of (3) form a

Received by the editor December 18, 1992 and, in revised form, April 13, 1993.
1991 *Mathematics Subject Classification.* Primary 11T06, 11Y16.
Key words and phrases. Polynomial factorization, finite fields of characteristic 2.

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

linear subspace $L(f)$ of the vector space $F[x]$ over the binary field $\mathbb{F}_2$. In [10], Niederreiter shows that the solutions of the differential equation (3) are given exactly by the 2^m polynomials h of the form

$$h = \frac{f}{b} b', \tag{4}$$

where $b \in F[x]$ is a monic factor of $g_1 \cdots g_m$, so that the $\mathbb{F}_2$-vector space $L(f)$ has dimension m. For $h \in L(f)$ it follows from (4) that

$$\gcd(f, h) = \frac{f}{b} \gcd(b, b') = \frac{f}{b},$$

and hence

$$\frac{f}{\gcd(f, h)} = b.$$

In view of these facts, two strategies to factor f suggest themselves.

In the first strategy, a solution polynomial h of (3) with $h \neq 0$ and $h \neq f'$ is determined. For this polynomial, $\gcd(f, h)$ is a nontrivial factor of f. One then applies the factorization algorithm to this nontrivial factor and its complementary factor of f and iterates.

In the second strategy, all 2^m solution polynomials h of (3) are calculated. The corresponding polynomials $f/\gcd(f, h)$ then produce all 2^m monic factors b of the squarefree part $g_1 \cdots g_m$ of f and, in particular, all irreducible factors of f.

Although only the first strategy leads to a polynomial-time algorithm, the second strategy is more efficient in most practical cases. In this paper we develop a third strategy. We shall show that the m polynomials of any basis of $L(f)$ are already sufficient to produce the irreducible factors $g_1, \ldots, g_m$ of f and with that the complete canonical factorization (1) of f. Our method leads to a polynomial-time algorithm, which is also very efficient for polynomials f of small degree. In contrast to the 2^m greatest common divisor calculations necessary in the second strategy, our method requires at most m^2 gcd calculations.

An important part of the Niederreiter algorithm is the actual computation of the polynomials $h \in F[x]$ which satisfy (3), or, to put it differently, the determination of a basis of $L(f)$. We shall discuss this matter only very briefly here and refer to the original papers [9, 10, 11].

It follows from (3) or (4) that any polynomial $h \in F[x]$ satisfying (3) must have degree $< d = \deg(f)$, so that we can set $h(x) = y_0 + y_1 x + \cdots + y_{d-1} x^{d-1}$ with all $y_j \in F$. Since F has characteristic 2, the first derivative $(fh)'$ is always a polynomial in x^2, and h^2 is also a polynomial in x^2 for all $h \in F[x]$. Thus, (3) holds if and only if the coefficients of x^{2j}, $0 \leq j \leq d-1$, agree on both sides. The comparison of coefficients yields a system of d equations for the unknowns $y_0, \ldots, y_{d-1} \in F$. This system of equations has the form

$$N(f)(y_0, \ldots, y_{d-1})^{\mathrm{T}} = (y_0^2, \ldots, y_{d-1}^2)^{\mathrm{T}}, \tag{5}$$

where the *Niederreiter matrix* $N(f)$ is a $d \times d$ matrix over F, obtained from

the coefficients of the polynomial f. If f is given by (2), then

$$(6)\qquad N(f) = \begin{pmatrix} f_1 & f_0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ f_3 & f_2 & f_1 & f_0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ f_5 & f_4 & f_3 & f_2 & f_1 & f_0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & f_d & f_{d-1} & f_{d-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & f_d \end{pmatrix}.$$

When the underlying perfect field F of characteristic 2 actually is a finite field of characteristic 2, then the system (5) can be turned into a homogeneous system of linear equations over the binary field $\mathbb{F}_2$.

If $F = \mathbb{F}_2$, we have $y_j^2 = y_j$ for all j, and the system (5) is already linear. In this case, we can write (5) in the form

$$(7)\qquad (N(f) - I_d)\mathbf{h}^{\mathrm{T}} = \mathbf{0},$$

with I_d the $d \times d$ identity matrix and $\mathbf{h} = (y_0, \dots, y_{d-1})$.

If $F = \mathbb{F}_q$ is a finite field of order $q = 2^t > 2$, a normal basis $B = \{\alpha, \alpha^2, \alpha^4, \dots, \alpha^{q/2}\}$ of $\mathbb{F}_q$ over $\mathbb{F}_2$ is used (see [10, 11]) to transform (5) into a $dt \times dt$ system of linear equations

$$(8)\qquad K_q(f, B)\mathbf{H}^{\mathrm{T}} = \mathbf{0}$$

over $\mathbb{F}_2$. Here, $K_q(f, B)$ is a $dt \times dt$ matrix over $\mathbb{F}_2$, and the solution vector $\mathbf{H} \in \mathbb{F}_2^{dt}$ determines $(y_0, \dots, y_{d-1}) \in \mathbb{F}_q^d$ of (5).

Since the dimension of the $\mathbb{F}_2$-vector space $L(f)$ is m, and since (3) is equivalent to (7) or (8), respectively, depending on whether $F = \mathbb{F}_2$ or $F = \mathbb{F}_q$, the number m of distinct monic irreducible factors of f in $F[x]$ is given by $m = d - \operatorname{rank}(N(f) - I_d)$ or $m = dt - \operatorname{rank}(K_q(f, B))$, respectively.

2. The refinement of the Niederreiter algorithm for characteristic 2

Let F again be an arbitrary perfect field of characteristic 2. Given a basis $\{h_1, \dots, h_m\}$ of $L(f)$, we form the polynomials $b_1, \dots, b_m$ by setting $b_i = f/\gcd(f, h_i)$ for $i = 1, \dots, m$. It follows from (4) that the $b_i \in F[x]$ are monic squarefree factors of f. By calculating gcd's, further monic squarefree factors of f are created from the polynomials $b_1, \dots, b_m$. These factors are listed in rows in a table of at most m rows.

The first row of the table consists solely of the polynomial b_1. The second row consists of two or three polynomials, namely of the nonconstant polynomials among

$$\gcd(b_2, b_1), \quad \frac{b_1}{\gcd(b_2, b_1)}, \quad \frac{b_2}{\gcd(b_2, b_1)}.$$

In the general step, the polynomials of the kth row, $1 < k \leq m$, are derived from b_k and the polynomials of the $(k-1)$st row in the following manner: Let $r_1, \dots, r_s$ be the polynomials in the $(k-1)$st row. Compute $d_j = \gcd(b_k, r_j)$ for $j = 1, \dots, s$. The kth row is then made up of the nonconstant polynomials

in the list

$$d_1, \ \frac{r_1}{d_1}, \ \ldots, \ d_s, \ \frac{r_s}{d_s}, \ \frac{b_k}{d_1 \cdots d_s}. \tag{9}$$

It is immediate that the polynomial array constructed in this way has the following properties:

(i) The polynomials in any single row are pairwise relatively prime monic squarefree factors of f;
(ii) The polynomial b_k occurs in the kth row, either in its original form or split up into some nontrivial factors;
(iii) Every polynomial in the $(k-1)$st row also occurs in the kth row, either in its original form or split up into two nontrivial factors.

It is clear from property (i) that the procedure can (and should) be stopped as soon as a row containing m polynomials has been reached. For, the m polynomials of that row must necessarily be the polynomials $g_1, \ldots, g_m$, the monic irreducible factors of f.

Usually, the kth row will contain more polynomials than the $(k-1)$st, but that is not guaranteed. For instance, when b_k happens to be identical with a polynomial r_j of the $(k-1)$st row, both rows will contain exactly the same polynomials. However, the procedure always succeeds, i.e., leads to a row with m polynomials, as is seen from the following theorem.

Theorem 1. *At the latest, the mth row contains the polynomials $g_1, \ldots, g_m$.*

Proof. According to (4), for each $i = 1, \ldots, m$, the polynomial $(f/g_i)g_i'$ is a solution of the differential equation (3), i.e., an element of $L(f)$. Since $\{h_1, \ldots, h_m\}$ is a basis of $L(f)$, we have

$$\frac{f}{g_i} g_i' = \alpha_1 h_1 + \cdots + \alpha_m h_m \qquad \text{for some } \alpha_1, \ldots, \alpha_m \in \mathbb{F}_2.$$

Dividing both sides by f and using (4), we get

$$\frac{g_i'}{g_i} = \alpha_1 \frac{b_1'}{b_1} + \cdots + \alpha_m \frac{b_m'}{b_m},$$

from which we can see that at least one of the polynomials $b_1, \ldots, b_m$ must be divisible by g_i. This fact together with (i), (ii), and (iii) implies that the product of all polynomials in the mth row is equal to $g_1 \cdots g_m$.

Now let us assume to the contrary that the mth row contains fewer than m polynomials. Then at least one polynomial in the mth row must be divisible by two different g_i's, say by g_1 and g_2. We claim that this implies that each of the polynomials b_k, $1 \le k \le m$, is either divisible by both g_1 and g_2 or relatively prime to $g_1 g_2$. For, suppose some b_k were divisible by only one of the two polynomials g_1 or g_2, but not by the other; then the kth row would contain a polynomial with the same property. But this, because of (iii), contradicts the appearance of a polynomial in the last row which is a multiple of $g_1 g_2$.

By the argument at the beginning of the proof, we have, in particular,

$$\frac{g_1'}{g_1} = \beta_1 \frac{b_1'}{b_1} + \cdots + \beta_m \frac{b_m'}{b_m} \qquad \text{for suitable } \beta_1, \ldots, \beta_m \in \mathbb{F}_2. \tag{10}$$

Now, for an arbitrary monic factor $b \in F[x]$ of $g_1 \cdots g_m$, it follows from the product rule that

$$\frac{b'}{b} = \sum \frac{g_j'}{g_j}, \tag{11}$$

where the sum is extended over all g_j's which divide b. Using (11), we see that the right-hand side of (10) can be viewed as a sum of terms g_j'/g_j. Because of the left-hand side of (10), the term g_1'/g_1 must occur an odd number of times in this sum, whereas all the other terms—in particular g_2'/g_2—occur an even number of times. But this contradicts the fact that each polynomial b_k is a multiple of $g_1 g_2$ or relatively prime to $g_1 g_2$. □

The Niederreiter algorithm for binary polynomials now takes the following form: Let $f \in \mathbb{F}_2[x]$ be the polynomial to be factored of degree $d \geq 1$.

Step 1. Set up the binary $d \times d$ matrix $N(f) - I_d$, where $N(f)$ is the Niederreiter matrix corresponding to f (see (6)) and I_d is the $d \times d$ identity matrix over $\mathbb{F}_2$. By a rank computation, we determine the number m of distinct irreducible factors of f in $\mathbb{F}_2[x]$, namely

$$m = d - \operatorname{rank}(N(f) - I_d).$$

Step 2. Solve the homogeneous system of linear equations

$$(N(f) - I_d)\mathbf{h}^{\mathrm{T}} = \mathbf{0}. \tag{12}$$

Each solution vector $\mathbf{h} = (y_0, y_1, \ldots, y_{d-1}) \in \mathbb{F}_2^d$ of (12) gives rise to a binary polynomial $y_0 + y_1 x + \cdots + y_{d-1} x^{d-1}$. From m arbitrary linearly independent (over $\mathbb{F}_2$) solution vectors $\mathbf{h}_1, \ldots, \mathbf{h}_m$ of (12), in this manner, m polynomials $h_1, \ldots, h_m \in \mathbb{F}_2[x]$ are obtained.

Step 3. Compute $b_1, \ldots, b_m \in \mathbb{F}_2[x]$ by

$$b_i = \frac{f}{\gcd(f, h_i)} \qquad \text{for } i = 1, \ldots, m.$$

Step 4. Set up a table of polynomials consisting of at most m rows as follows. The first row contains the polynomial b_1. The other rows are defined inductively. If the $(k-1)$st row contains the nonconstant polynomials $r_1, \ldots, r_s$ ($1 < k \leq m$, $1 \leq s < m$), then compute the polynomials

$$d_1, \quad \frac{r_1}{d_1}, \quad \ldots, \quad d_s, \quad \frac{r_s}{d_s}, \quad c_{s+1} \tag{13}$$

with $c_1 = b_k$, $d_j = \gcd(c_j, r_j)$ and $c_{j+1} = c_j/d_j$ for $j = 1, \ldots, s$. Remove (if any) all constant polynomials from the list (13); the remaining polynomials form the kth row. This process is continued until a row with m nonconstant polynomials is obtained. This may be the mth row or an earlier one. The polynomials of that row are then the distinct irreducible factors of f in $\mathbb{F}_2[x]$.

We note that the method described in Step 4 to compute the kth row from the $(k-1)$st is equivalent to the method described earlier, i.e., the polynomials d_j in (13) are the same as in (9), and $c_{s+1} = b_k/(d_1 \cdots d_s)$.

If the underlying field F is the finite field of order $q = 2^t > 2$, in the factorization algorithm of Niederreiter one first has to set up the $dt \times dt$ matrix

$K_q(f, B)$ over $\mathbb{F}_2$, where $d \geq 1$ is again the degree of $f \in F[x]$. The number m of distinct monic irreducible factors of f in $F[x]$ is, as mentioned above, obtained from $m = dt - \operatorname{rank}(K_q(f, B))$. Next, the homogeneous system of linear equations (8) over $\mathbb{F}_2$ has to be solved. Any m linearly independent (over $\mathbb{F}_2$) solution vectors $\mathbf{H} \in \mathbb{F}_2^{dt}$ of (8) yield the polynomials $h_1, \ldots, h_m \in F[x]$. One now has to apply the method described in Steps 3 and 4 of the binary Niederreiter algorithm to $h_1, \ldots, h_m$, which yields all monic irreducible factors of f in $F[x]$.

If the underlying perfect field F of characteristic 2 is infinite, one has to solve the nonlinear system of equations (5) over F. Once a basis of $L(f)$ is found, the method described in Steps 3 and 4 above can again be used to get all monic irreducible factors of f in $F[x]$.

3. Complexity analysis

Throughout this section the underlying field is a finite field $\mathbb{F}_q$ of order $q = 2^t \geq 2$. Suppose m polynomials $h_1, \ldots, h_m$ forming a basis of $L(f)$ have already been determined. In order to produce from these the monic irreducible factors $g_1, \ldots, g_m$ of f in the manner described in §2, we need only to compute gcd's and perform divisions, in either case of polynomials in $\mathbb{F}_q[x]$ of degree $\leq d = \deg(f)$. A rough estimation shows that at most m^2 gcd's have to be calculated and at most m^2 divisions have to be performed.

The gcd of two polynomials in $\mathbb{F}_q[x]$ of degree $\leq d$ can be calculated with $O(M_q(d) \log d)$ arithmetic operations in $\mathbb{F}_q$, where $M_q(d)$ is the arithmetic complexity of multiplying two polynomials in $\mathbb{F}_q[x]$ of degree $\leq d$ (see [1, p. 308, Theorem 8.19]). The arithmetic complexity of dividing two polynomials in $\mathbb{F}_q[x]$ of degree $\leq d$ has the same order of magnitude as $M_q(d)$ (see [1, p. 288, Theorem 8.7]). Therefore, the computation of the polynomials $g_1, \ldots, g_m$ from the polynomials $h_1, \ldots, h_m$ requires $O(m^2 M_q(d) \log d)$ arithmetic operations in $\mathbb{F}_q$. The function $M_q(d)$ is $O(d(\log d) \log\log d)$ according to Cantor and Kaltofen [2], and a somewhat better estimation may be derived from Grigoriev [4] or Lempel et al. [6].

We now estimate the total cost of the binary Niederreiter algorithm, that is, for the case $q = 2$. Since the matrix $N(f)$ can be read off immediately from the coefficients of the polynomial f, there is no setup cost for the matrix $N(f) - I_d$ in (12). The system (12) itself, being a $d \times d$ system of linear equations over $\mathbb{F}_2$, can be solved with $O(d^\omega)$ arithmetic operations in $\mathbb{F}_2$, where $\omega < 2.38$ is the exponent of fast matrix multiplication. The cost of Steps 3 and 4 of the algorithm has already been estimated above. Hence, the following theorem is proved.

Theorem 2. *The total cost of calculating all m irreducible factors of a binary polynomial of degree d by the Niederreiter algorithm is $O(d^\omega + m^2 M_2(d) \log d)$ arithmetic operations in $\mathbb{F}_2$, where $\omega < 2.38$ is the exponent of fast matrix multiplication and $M_2(d)$ is the arithmetic complexity of multiplying two binary polynomials of degree $\leq d$.*

There is some hope for the binary Niederreiter algorithm to be further accelerated, owing to the special form of the matrix $N(f)$. In [10] Niederreiter posed the problem of developing a method for solving (12) with $O(d^2)$ arithmetic operations in $\mathbb{F}_2$. If this succeeded, the binary Niederreiter algorithm

would be an $O(d^2)$ algorithm for random polynomials, since the average order of magnitude of the number m of distinct irreducible factors of f is $\log d$ according to [7, pp. 239–241].

If the underlying field $\mathbb{F}_q$ is of order $q = 2^t > 2$, one first of all has to set up the matrix $K_q(f, B)$. It was shown in [10] that the setup cost for the matrix $K_q(f, B)$ is $O(dt^3)$ arithmetic operations in the binary field $\mathbb{F}_2$. The $dt \times dt$ system of linear equations (8) can be solved with $O(d^\omega t^\omega)$ arithmetic operations in the binary field $\mathbb{F}_2$, after which the polynomials $h_1, \ldots, h_m \in \mathbb{F}_q[x]$ are available. We summarize the total cost of computation in the following theorem.

Theorem 3. *The total cost of calculating all m monic irreducible factors of a polynomial in $\mathbb{F}_q[x]$, $q = 2^t > 2$, of degree d by the Niederreiter algorithm is $O(dt^3 + d^\omega t^\omega)$ arithmetic operations in the binary field $\mathbb{F}_2$ plus $O(m^2 M_q(d) \log d)$ arithmetic operations in $\mathbb{F}_q$. Here $\omega < 2.38$ is the exponent of fast matrix multiplication and $M_q(d)$ is the arithmetic complexity of multiplying two polynomials in $\mathbb{F}_q[x]$ of degree $\leq d$.*

In order to express the total cost of computation in terms of arithmetic operations in the binary field $\mathbb{F}_2$, we use the fact that any arithmetic operation in the finite field of order 2^t can be accomplished with at most $O(t(\log t)\log\log t)$ arithmetic operations in $\mathbb{F}_2$. The latter follows from the theorem in Cantor and Kaltofen [2]. Using also the estimation for $M_q(d)$ given earlier, we can restate Theorem 3 in the following manner.

Corollary. *With the notation in Theorem 3, the complexity of the Niederreiter algorithm is*

$$O(dt^3 + d^\omega t^\omega + m^2 d(\log d)^2(\log\log d)t(\log t)\log\log t)$$

arithmetic operations in the binary field $\mathbb{F}_2$. In particular, the algorithm runs in polynomial time.

Example. We illustrate the binary Niederreiter algorithm with the polynomial $f(x) = x^{11} + x^8 + x^5 + x^4 + 1 \in \mathbb{F}_2[x]$. The matrix $N(f)$ is

$$N(f) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

By means of elementary row operations, we reduce the matrix $N(f) - I_{11}$ to

echelon form. This yields the matrix

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

We now have $\operatorname{rank}(N(f) - I_{11}) = \operatorname{rank}(A) = 8$, so that the canonical factorization of f over $\mathbb{F}_2$ contains $11 - 8 = 3$ distinct irreducible factors. Since we have used only elementary row operations, the matrices $N(f) - I_{11}$ and A have the same null space. Therefore, we can solve the linear system $A\mathbf{h}^{\mathrm{T}} = \mathbf{0}$ instead of (12). This yields three linearly independent solution vectors

$$\mathbf{h}_1 = (0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1),$$
$$\mathbf{h}_2 = (1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0),$$
$$\mathbf{h}_3 = (0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0).$$

The corresponding polynomials are given by

$$h_1(x) = x^{10}+x^4, \quad h_2(x) = x^9+x^8+x^3+x+1, \quad h_3(x) = x^9+x^8+x^7+x^5+x^2,$$

from which we obtain the polynomials

$$b_1(x) = x^7 + x^5 + x^4 + x^2 + 1, \quad b_2(x) = x^2 + x + 1, \quad b_3(x) = x^4 + x^3 + 1.$$

The table of polynomials in Step 4 is

$$\begin{aligned} \text{1st row:} \quad & x^7 + x^5 + x^4 + x^2 + 1 \\ \text{2nd row:} \quad & x^7 + x^5 + x^4 + x^2 + 1, \quad x^2 + x + 1 \\ \text{3rd row:} \quad & x^4 + x^3 + 1, \quad x^3 + x^2 + 1, \quad x^2 + x + 1 \end{aligned}$$

Since $\deg(f) = 11$ the canonical factorization of f over $\mathbb{F}_2$ must be

$$f(x) = (x^4 + x^3 + 1)(x^3 + x^2 + 1)(x^2 + x + 1)^2.$$

ACKNOWLEDGMENT

The author would like to express special thanks to Mag. Philip Bajo and to Professor Harald Niederreiter for valuable comments and suggestions. The useful remarks of the referee are also appreciated.

BIBLIOGRAPHY

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, Reading, Mass., 1974.
2. D. G. Cantor and E. Kaltofen, *On fast multiplication of polynomials over arbitrary algebras*, Acta Inform. **28** (1991), 693–701.

3. P. Fleischmann, *Connections between the algorithms of Berlekamp and Niederreiter for factoring polynomials over* $\mathbb{F}_q$, Linear Algebra Appl. **192** (1993), 101–108.
4. D. Yu. Grigoriev, *Multiplicative complexity of a pair of bilinear forms and of the polynomial multiplication*, Mathematical Foundations of Computer Science 1978 (J. Winkowski, ed.), Lecture Notes in Comput. Sci., vol. 64, Springer-Verlag, Berlin, 1978, pp. 250–256.
5. T. C. Y. Lee and S. A. Vanstone, *Subspaces and polynomial factorizations over finite fields*, Applicable Algebra in Engrg. Comm. Comp. (to appear).
6. A. Lempel, G. Seroussi, and S. Winograd, *On the complexity of multiplication in finite fields*, Theoret. Comput. Sci. **22** (1983), 285–296.
7. M. Mignotte, *Mathematics for computer algebra*, Springer-Verlag, New York, 1992.
8. V. S. Miller, *On the factorization method of Niederreiter*, IBM T. J. Watson Research Center, Yorktown Heights, N.Y., 1992, preprint.
9. H. Niederreiter, *A new efficient factorization algorithm for polynomials over small finite fields*, Applicable Algebra in Engrg. Comm. Comp. **4** (1993), 81–87.
10. ———, *Factorization of polynomials and some linear-algebra problems over finite fields*, Linear Algebra Appl. **192** (1993), 301–328.
11. ———, *Factoring polynomials over finite fields using differential equations and normal bases*, Math. Comp. **62** (1994), 819–830.
12. H. Niederreiter and R. Göttfert, *Factorization of polynomials over finite fields and characteristic sequences*, J. Symbolic Comput. (to appear).

KENYONGASSE 20/30, A-1070 VIENNA, AUSTRIA
E-mail address: goet@qiinfo.oeaw.ac.at