Connections between the Algorithms of Berlekamp and Niederreiter for factoring Polynomials over $\mathbf{F}_q$

Peter Fleischmann
Institut für
Experimentelle Mathematik
Universität Essen

Introduction

Factorizing polynomials over the finite field $\mathbf{F}_q$ is an important task in pure and applied algebra. Using standard methods it always comes down to finding non - trivial factors of a separable polynomial, i.e. a polynomial f with no multiple roots in the algebraic closure $\overline{\mathbf{F}}_q$. In this case the classical algorithm of Berlekamp [B] tranfers this problem to linear algebra by reducing it to the solution of linear equations.

Recently in [N1],[N2], Niederreiter proposed a different linearization technique, based on the analysis of certain differential equations in the field of rational functions $\mathbf{F}_q(X)$.

His method has the following practical advantages:

Unlike Berlekamp's method it works for polynomials with multiple roots in the same way as for separable ones (see [N2]). It also reduces the cost of setting up the linear equations to be solved by providing generic patterns depending only on the degree of f. In particular in characteristic 2 the algorithm of Niederreiter preserves sparsity in the sense that a sparse polynomial f leads to a sparse system of linear equations. These can be solved much faster than general systems using special techniques. Notice that Berlekamp's algorithm does not necessarily preserve sparsity.

Supppose that f is squarefree. In this case the solution space of Berlekamp's algorithm has a nice and useful algebraic interpretation. In fact several modifications and accelerations, mostly important for large fields, have been proposed, that essentially use Berlekamp's solutions (see section 3).

In this note we will show that after an easy transformation, the solution space of Niederreiter's algorithm coincides with the one of Berlekamp's (see corollary in section 2). Hence an implementation of a combined procedure will inherit the advantages of both algorithms.

1. The decomposition algebra of a polynomial

Let $\mathbf{F}_q$ be the finite field with q elements. Throughout this discussion we fix a monic polynomial $f \in \mathbf{F}_q[X]$ of degree $d \geq 1$. We will assume f to be squarefree or, in other words, the greatest common divisor $g.c.d.(f, f')$ of f and its derivative f' will always be one. In this case the canonical factorization of f over $\mathbf{F}_q$ looks like

$$f = g_1 \cdot g_2 \cdot \ldots \cdot g_m \tag{1}$$

with distinct irreducible monic polynomials $g_i \in \mathbf{F}_q[X]$.

Notice that since f is squarefree, the algebra $\mathcal{A}_f := \mathbf{F}_q[X]/(f)$ is isomorphic to the direct sum $\oplus_{i=1}^m \mathbf{F}_q[X]/(g_i)$ and each summand $\mathbf{F}_q[X]/(g_i)$ is isomorphic to an extension field, whose degree over $\mathbf{F}_q$ coincides with the degree of the polynomial g_i. In particular we see that the algebra $\mathcal{A}_f$ is semisimple.

Due to the chinese remainder theorem there exists for each $i \in \{1, 2, ..., m\}$ a unique polynomial $e_i \in \mathbf{F}_q[X]$ such that

a) the degree of e_i is less then d and

b) $e_i \equiv 1 \ mod \ g_i$ and $e_i \equiv 0 \ mod \ g_j$ for all $j \in \{1, 2, ..., m\} \backslash \{i\}$.

The corresponding residue classes form the set $\{e_1 + (f), e_2 + (f), ..., e_m + (f)\}$ of primitive orthogonal idempotents in $\mathcal{A}_f$, which is unique since this algebra is commutative.

A linear combination $v = \sum_{i=1}^m c_i \cdot e_i$ with $c_i \in \mathbf{F}_q$ is a unit modulo f, if and only if all the coefficients c_i are different from zero. A non - zero linear combination v which does not give a unit in $\mathcal{A}_f$, clearly yields a non - trivial divisor of f if we compute $g.c.d.(v, f)$. Hence the problem of finding non - trivial factors and, eventually, the decomposition (1) of f, can be stated as follows:

Goal: Find in an efficient way **non - units** inside the **subalgebra**

$$\mathcal{D}_f := < e_1 + (f), e_2 + (f), ..., e_m + (f) >_{\mathbf{F}_q} \quad \leq \quad \mathcal{A}_f.$$

For this reason we like to call $\mathcal{D}_f$ the **decomposition algebra of** f.

In the theory of Berlekamp's algorithm one has the notion of f - **reducing polynomials**. These are polynomials $h \in \mathbf{F}_q[X]$ with degree less then d, satisfying the equation $f = \prod_{c \in \mathbf{F}_q} g.c.d.(h - c, f)$. They can be characterized algebraically as follows:

Lemma: *Let $h \in \mathbf{F}_q[X]$ be a polynomial with degree less then d, then the following are equivalent:*

(1) h is an f - reducing polynomial;

(2) $h^q - h \equiv 0 \ mod \ (f)$;

(3) h is an element of $\mathcal{D}_f$.

Proof: The equivalence of (1) and (2) is the classical starting point of Berlekamp's algorithm. It follows from the factorization

$$X^q - X = \prod_{c \in \mathbf{F}_q} X - c$$

which yields: $h^q - h = (X^q - X)_{X=h} = (\prod_{c\in\mathbf{F}_q} X - c)_{X=h} = \prod_{c\in\mathbf{F}_q} h - c$. Hence f divides $h^q - h$ if and only if $f = \prod_{c\in\mathbf{F}_q} g.c.d.(h-c, f)$.

Now suppose that h is f - reducing. Then for each $j \in \{1, 2, ..., m\}$ there is a unique $c_j \in \mathbf{F}_q$ such that g_j divides $g.c.d.(h - c_j, f)$. Hence $\sum_{i=1}^m c_i \cdot e_i \equiv c_j \equiv h \ mod \ (g_j)$ for all j. So f divides $h - \sum_{i=1}^m c_i \cdot e_i$, and since the degree of h is less than d, we conclude $h = \sum_{i=1}^m c_i \cdot e_i$. On the other hand each $\mathbf{F}_q$ - linear combination $\sum_{i=1}^m c_i \cdot e_i$ satisfies $(\sum_{i=1}^m c_i \cdot e_i)^q \equiv \sum_{i=1}^m (c_i)^q \cdot (e_i)^q \equiv \sum_{i=1}^m c_i \cdot e_i \ mod \ (f)$. ♣

2. The algorithms of Berlekamp and Niederreiter

By Berlekamp's algorithm, finding a basis for the decomposition algebra $\mathcal{D}_f$ is reduced to the problem of computing the null-space of a certain $d \times d$ - matrix $\mathcal{Q} - E$ over $\mathbf{F}_q$ (E is the identity matrix):

Observe that each polynomial $g[X] \in \mathbf{F}_q[X]$ satisfies $g[X]^q = g[X^q]$. If $\mathcal{Q} = (q_{k,l})_{k,l\in\{0,1,\ldots,d-1\}}$ is defined by $x^{qk} \equiv \sum_{l=0}^{d-1} q_{k,l} \cdot x^l \ mod \ (f)$ then one can see immediately that the polynomial $h = \sum_{l=0}^{d-1} h_l \cdot x^l$ is in $\mathcal{D}_f$ if and only if its row vector of coefficients $\mathbf{h} = (h_0, h_1, ..., h_{d-1})$ satisfies $\mathbf{h} \circ \mathcal{Q} = \mathbf{h}$, or, if $\mathbf{h}$ is contained in the (left) null - space of $\mathcal{Q} - E$.

By the discussion above this null - space must have dimension m, in particular one can recognize irreducible polynomials if this dimension is one.

In [N1] Niederreiter proposed a new algorithm for factorizing polynomials over $\mathbf{F}_p$. Later he generalized this procedure to general finite fields $\mathbf{F}_q$ (see [N2]). His method is also based on the computation of the kernel of a $d \times d$ - matrix $\mathcal{N} - E$ over $\mathbf{F}_q$, which on the first glance does not seem to be related at all to the matrix $\mathcal{Q} - E$. Let us give a sketchy description of Niederreiter's algorithm:

Starting point in case $q = p$ is the differential equation

$$y^{(p-1)} + y^p = 0 \tag{2}$$

of order $p-1$ in the rational function field $\mathbf{F}_p(x)$. In the general case of $\mathbf{F}_q$ this is replaced by the differential equation

$$H^{(q-1)}(y) - y^q = 0. \tag{3}$$

Here $H^{(k)}$ is the **Hasse - Teichmüller - derivative** of formal Laurent series over $\mathbf{F}_q$ in x^{-1} which is defined by:

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

The solutions of (2) and (3) form an $\mathbf{F}_q$ - subspace of $\mathbf{F}_q(X)$ consisting of functions of degree ≤ -1. Niederreiter proved that the subspace of all those solutions having the form $y = h/f$ with fixed denominator f and $h \in \mathbf{F}_q[X]$ coincides with the set

$$\{ \ y = \sum_{i=1}^{m} c_i \cdot \frac{g_i{}'}{g_i} \ | \ c_1, ..., c_m \in \mathbf{F}_q \ \}.$$

For solutions of this type, the equations (2) and (3) can be rewritten in the form

$$f^q \cdot H^{(q-1)}(\frac{h}{f}) = h^q \tag{4}$$

with polynomials of degree $\leq (d-1) \cdot q$ on both sides. Since these are polynomials in x^q as well, a comparison of coefficients shows the equivalence of (2) and (3) with the equation

$$(\mathcal{N} - E)(\mathbf{h})^t = 0 \qquad (5),$$

where $\mathbf{h} = (h_0, ..., h_{d-1})$ is the row vector consisting of the coefficients of h and $\mathcal{N}$ is a $d \times d$ matrix over $\mathbf{F}_q$ depending on f and the quotient rules for the usual derivation if $q = p$ or for $H^{(q-1)}$ in general.

Since any solution h satisfies $h = \sum_{i=1}^{m} c_i \cdot g_i{}' \prod_{j \neq i} g_j \equiv c_{i_0} \cdot g_{i_0}{}' \prod_{j \neq i_0} g_j \ mod \ g_{i_0}$, building $g.c.d.(h, f)$ will yield non - trivial factors of f whenever at least one but not all of the c_i's are zero. We define $\mathcal{N}_f := \{h \in \mathbf{F}_q[X] \mid deg(h) < deg(f), \ f^q H^{(q-1)}(\frac{h}{f}) = h^q \ \}$ i.e. the $\mathbf{F}_q$ - space of solutions of Niederreiter's algorithm.

An advantage of this method is that a generic pattern $\tilde{\mathcal{N}}$, valid for a given degree d can be produced in a precomputation using the quotient rules. From this the actual matrix $\mathcal{N}$ will be obtained by inserting the coefficients of f (see [N1] for some examples). Unlike this, the matrix $\mathcal{Q}$ of Berlekamp's algorithm has to be computed recursively and independently for each polynomial f.

The following observation is the key to the connection between Berlekamp's and Niederreiter's methods:

Proposition: *Let K be an arbitrary field and $g, h \in K[X]$ polynomials such that $deg \ h < deg \ g$. Then the usual derivative and the Hasse - Teichmüller derivative satisfy the following congruences:*

$$(\frac{h}{g})^{(k)} \cdot g^{k+1} \ \equiv \ (-1)^k \cdot k! \cdot h \cdot (g')^k \quad mod \ g; \tag{6}$$

$$H^{(k)}(\frac{h}{g}) \cdot g^{k+1} \ \equiv \ (-1)^k \cdot h \cdot (g')^k \quad mod \ g. \tag{7}$$

Proof: Notice that $()^{(k)} = k! \cdot H^{(k)}()$, so the second equation implies the first. It can also be derived directly using the quotient rule and induction on k. For the second equation we use the partial fraction decomposition of $\frac{h}{g}$ and a lemma of Göttfert on the Hasse - Teichmüller derivative, stated in [N2].

By Göttfert's lemma we have $H^{(k)}((x-a)^{-r}) = \binom{-r}{k} \cdot (x-a)^{-r-k}$. Suppose $g = \prod_{\beta \in R}(x-\beta)^{e(\beta)}$ is the factorization of g over an algebraic closure $\overline{K}$ and R the set of roots of g. Then the partial fraction decomposition of h/g is of the form

$$\frac{h}{g} = \sum_{\beta \in R} \sum_{r=1}^{e(\beta)} \frac{\gamma(\beta, r)}{(x-\beta)^r},$$

with $\gamma(\beta, r) \in \overline{K}$. Hence $h = \sum_{\beta \in R} \sum_{r=1}^{e(\beta)} \gamma(\beta, r) \cdot (x-\beta)^{e(\beta)-r} \cdot \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{e(\alpha)}$.

Now we get: $H^{(k)}(h/g) \cdot g^{k+1} =$

$$\sum_{\beta \in R} \sum_{r=1}^{e(\beta)} \binom{-r}{k} \cdot \frac{\gamma(\beta, r)}{(x-\beta)^{r+k}} \cdot \prod_{\alpha \in R} (x-\alpha)^{e(\alpha) \cdot (k+1)}$$

$$= \sum_{\beta \in R} \sum_{r=1}^{e(\beta)} \binom{-r}{k} \cdot \gamma(\beta, r) \cdot (x-\beta)^{k(e(\beta)-1)+e(\beta)-r} \cdot \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{e(\alpha) \cdot (k+1)}$$

$$\equiv \sum_{r=1}^{e(\beta)} \binom{-r}{k} \cdot \gamma(\beta, r) \cdot (x-\beta)^{k(e(\beta)-1)+e(\beta)-r} \cdot \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{e(\alpha) \cdot (k+1)} \ mod \ (x-\beta)^{e(\beta)}.$$

This is equal to $0 \ mod \ (x-\beta)^{e(\beta)}$ if k and $e(\beta)$ are greater than one, since then $k \cdot (e(\beta)-1) - r = k \cdot e(\beta) - k - r \geq k + e(\beta) - k - r \geq 0$. If $k = 1$ it is equal to $-e(\beta) \cdot \gamma(\beta, e(\beta)) \cdot (x-\beta)^{e(\beta)-1} \times *$ and if $e(\beta) = 1$ it equals $\binom{-1}{k} \cdot \gamma(\beta, 1) \times *$ modulo $(x-\beta)^{e(\beta)}$ with $* = (\prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{e(\alpha)})^{k+1}$. Notice also that $\binom{-1}{k} = (-1)^k$.

In a similar way we get for the right hand side: $h \cdot (f')^k \equiv$

$$\left(\sum_{r=1}^{e(\beta)} \gamma(\beta, r) \cdot (x-\beta)^{e(\beta)-r} \cdot \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{e(\alpha)} \right) \times$$

$$\left(e(\beta)(x-\beta)^{e(\beta)-1} \cdot \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{e(\alpha)} \right)^k \ mod \ (x-\beta)^{e(\beta)},$$

which is equal to $0 \ mod \ (x-\beta)^{e(\beta)}$ if k and $e(\beta)$ are greater than one. If $k = 1$ it is equal to $e(\beta) \cdot \gamma(\beta, e(\beta)) \cdot (x-\beta)^{e(\beta)-1} \times *$ and if $e(\beta) = 1$ it equals $\gamma(\beta, 1) \times *$ modulo $(x-\beta)^{e(\beta)}$.♣

Corollary: *Let $f \in \mathbf{F}_q[X]$ be a squarefree polynomial and let $a, b \in \mathbf{F}_q[X]$ be polynomials satisfying the equation $1 = a \cdot f' + b \cdot f$. Then multiplication by a and reduction modulo f yields an isomorphism of vectorspaces:*

$$\mathcal{N}_f \longrightarrow \mathcal{D}_f, \ h \mapsto a \cdot h + (f).$$

Proof: The polynomial h is in $\mathcal{N}_f$ if and only if it has degree less then d and satisfies $H^{(q-1)}(h/f) \cdot f^q = h^q$. For each of these h we get: $(a \cdot h)^q \equiv a^q \cdot H^{(q-1)}(h/f) \cdot f^q \equiv a^q \cdot h \cdot (f')^{q-1} \equiv a \cdot h \cdot (af')^{q-1} \equiv (a \cdot h) \; mod \; (f)$. Since multiplication by a is invertible modulo (f), we get an isomorphism of $\mathbf{F}_q$ - vectorspaces. ♣

It is clear that polynomials $a, b \in \mathbf{F}_q[X]$ with $1 = a \cdot f' + b \cdot f$ can be obtained in the beginning of the procedure as a byproduct of testing or reducing to separability by euclid's algorithm.

3. Concluding remarks

Since the algorithms of Berlekamp and Niederreiter both produce a basis $\mathcal{B}$ for the decomposition algebra $\mathcal{D}_f$, we want to recall briefly its use in actual implementations:
If the field $\mathbf{F}_q$ is small one simply can compute all greatest common divisors $g.c.d.(b + c, g)$ for all $c \in \mathbf{F}_q$, $1 \neq b \in \mathcal{B}$ and all proper factors g of f found in previous steps, to get eventually a full factorization.

If q is large one can use a method of Zassenhaus to reduce the number of elements $c \in \mathbf{F}_q$ for which $g.c.d.(b + c, f)$ has to be computed:
Notice that the algebra $\mathcal{D}_f$ is isomorphic to m - copies of $\mathbf{F}_q$. Hence for each $b \in \mathcal{B}$ the minimal-polynomial m_b of the element $b + (f) \in \mathcal{D}_f$ over $\mathbf{F}_q$ splits into linear factors. The zeroes z of m_b are exactly those elements in $\mathbf{F}_q$ such that $b - z + (f)$ is not a unit in $\mathcal{D}_f$, thus yielding a non - trivial divisor of f. This reduces the general factorization problem to the question how to find zeroes of polynomials over $\mathbf{F}_q$, which are known to split into linear factors.

For large fields of odd characteristic one can also adapt a very efficient probabilistic method, described by Cantor and Zassenhaus [CZ], to construct non-units in $\mathcal{D}_f$:
Suppose we have got a unit $b \in \mathcal{D}_f$, then the $(q-1)/2$ - th power $\tilde{b}$ of b modulo f can be computed efficiently using repeated squares. As a linear combination of the idempotents $e_i + (f)$, all coefficients of $\tilde{b}$ are 1 or -1. Except in the case where $\tilde{b}$ itself is 1 or -1 modulo f, we obtain the two coprime non units $\tilde{b} + 1$ and $\tilde{b} - 1$ in $\mathcal{D}_f$. (In fact $1/2(\tilde{b} + 1)$ and $1/2(\tilde{b} - 1)$ are orthogonal idempotents!) Out of the $q^m - 1$ non - zero elements of $\mathcal{D}_f$, only $2 \times (\frac{(q-1)}{2})^m$ will lead to ± 1 modulo (f). Thus by choosing elements in $\mathcal{D}_f$ at random we succeed with a probability of $1 - (\frac{1}{2})^{m-1} \cdot \frac{(q-1)^m}{q^m - 1} \geq 1/2$ for $m > 1$. (This procedure can be modified to work in the case were q is even (see [CZ])).

Having found a nontrivial factorization $f = f_1 \cdot f_2$, we know $\mathcal{D}_{f_i} = \mathcal{D}_f + (f_i)$ for $i = 1$ or 2. Thus to go on finding factors of f_i, we do not have to compute $\mathcal{D}_{f_i}$ by solving linear equations. Instead reducing the elements of $\mathcal{D}_f$ modulo f_i gives the same result (with linear dependencies). In particular f_i shows to be irreducible if $\mathcal{D}_f$ consists of constants modulo f_i.
Notice that from this point on, calculations modulo different f_i's can be carried out independently thus giving opportunity to parallel processing.

References

[B] **E.R.Berlekamp**, Factoring polynomials over finite fields, *Bell System Tech. J.* **46**, (1967), 1853 - 1859

[CZ] **D.G.Cantor, H.Zassenhaus**, A new algorithm for factoring polynomials over finite fields, *Math. of Comp.* **Vol.36, No.154**, (1981), 587 - 592

[N1] **H.Niederreiter**, A new efficient factorization algorithm for polynomials over small finite fields, to appear in *Applicable Algebra in Engrg. Comm. Comp.*

[N2] **H.Niederreiter**, Factorization of polynomials and some linear algebra problems over finite fields, to appear in *Linear Algebra Appl.*

revised version January 1993