On key equation

Article *in* IEEE Transactions on Information Theory · October 1995

DOI: 10.1109/18.412677 · Source: IEEE Xplore

1 author:

Patrick Fitzpatrick
University College Cork
69 PUBLICATIONS **728** CITATIONS

SEE PROFILE

On the Key Equation

Patrick Fitzpatrick

***Abstract*— We consider the set $\boldsymbol{M} = \{(a, b): a \equiv bh \bmod x^{2t}\}$ of all solutions of the key equation for alternant codes, where h is the syndrome polynomial. In decoding these codes a particular solution $(\omega, \sigma) \in \boldsymbol{M}$ is sought, subject to ω and σ being relatively prime and satisfying certain degree conditions. We prove that these requirements specify (ω, σ) uniquely as the *minimal element* of $\boldsymbol{M}$ (analogous to the monic polynomial of minimal degree generating an ideal of $\boldsymbol{F}[x]$) with respect to a certain term order and that, as such, (ω, σ) may be determined from an appropriate Gröbner basis of $\boldsymbol{M}$. Motivated by this and other variations of the key equation (such as that appropriate to errors-and-erasures decoding) we derive a general algorithm for solving the congruence $a \equiv bg \bmod x^n$ for a range of term orders defined by the conditions on the particular solution required. Our techniques provide a unified approach to the solution of these key equations.**

***Index Terms*— Gröbner basis, alternant code, decoding algorithms, errors-and-erasures.**

I. Introduction

THE CENTRAL computation in decoding alternant codes (including BCH, RS, and 1-variable Goppa codes) is the determination of polynomials $a, b \in \boldsymbol{A} = \boldsymbol{F}[x]$, $\boldsymbol{F}$ a finite field, satisfying the key equation which we write in the form of a congruence as

$$\omega \equiv \sigma h \bmod x^{2t}. \tag{1}$$

Here h is the syndrome polynomial and σ, ω represent the error locator and error evaluator polynomials, respectively. If δf denotes the degree of the polynomial f then $\delta h \leq 2t - 1$, and σ, ω are relatively prime polynomials with $\delta\omega < \delta\sigma \leq t$.

The following names are associated with algorithms that have been developed for the solution of the key equation: Peterson–Gorenstein–Zierler [3], [22], [32], [33], Berlekamp–Massey [2]–[4], [26], and Sugiyama *et al.*, who developed the technique based on the extended Euclidean algorithm [4], [27], [28], [35]. We refer to these algorithms as PGZ, BM, and E, respectively; all of them can be used in the decoding of alternant codes [3], [23], [31]. Moreover, similar congruences also arise in a variety of different contexts including errors-and-erasures decoding, linear recurring sequences, continued fractions, Hankel and Toeplitz linear systems, Padé approximation, polynomial inversion, and so on ([4], [5], [19], [29], [30], [36], for example). This has prompted many authors to study the interrelationships among these algorithms and particular effort has been devoted to the comparison between BM and E [7], [12], [13], [20], [25].

Manuscript received October 18, 1993; revised November 18, 1994. A summary of an early version of this paper is contained in [15]; the solution of the errors-and-erasures problem was outlined in [16].

The author is with the Department of Mathematics, University College, Cork, Ireland.

IEEE Log Number 9413876.

Congruence (1) may be viewed as an instance of the more general

$$a \equiv bg \bmod x^n \tag{2}$$

where $\delta g \leq n - 1$ and we want to find a particular solution (a, b), that we call the *required solution*, satisfying

$$\delta a \leq \ell, \quad \delta b \leq m, \quad \ell + m < n$$

$$a \text{ and } b \text{ relatively prime} \tag{3}$$

for some nonnegative integers ℓ, m. Other special cases of (2) include Berlekamp's (original) key equation

$$\hat{\omega} \equiv \sigma(1 + xh) \bmod x^{2t+1}$$

[2], which is itself a special case of the errors-and-erasures congruence

$$\lambda \equiv \mu(1 + xH) \bmod x^{2t+1}$$

where λ, μ are the error evaluator and error locator polynomials, respectively, and H is the "modified" syndrome polynomial [2], [3], [16], [34].

In this paper we use the theory of Gröbner bases to develop techniques that unify these different manifestations of the key equation and lead to new algorithms corresponding to PGZ, BM, and E. For a specific choice of the parameter $r = \ell - m$, two of our algorithms are computationally equivalent to PGZ and E, while the third is computationally better than BM.

We denote by

$$\boldsymbol{M} = \{(a, b): a \equiv bg \bmod x^n\} \subseteq \boldsymbol{A}^2 = \boldsymbol{A} \times \boldsymbol{A}$$

the solution set of (2). It is clear that $\boldsymbol{M}$ is closed under addition ($a \equiv bh$ and $c \equiv dh$ imply $a + c \equiv (b + d)h$, i.e., $(a, b), (c, d) \in \boldsymbol{M}$ imply $(a + c, b + d) \in \boldsymbol{M}$) and multiplication by polynomials ($a \equiv bh$ and $f \in \boldsymbol{A}$ imply $fa \equiv (fb)h$, i.e., $(a, b) \in \boldsymbol{M}$ implies $f(a, b) = (fa, fb) \in \boldsymbol{M}$). By definition, this means that $\boldsymbol{M}$ is a *submodule* of $\boldsymbol{A}^2$ (corresponding structurally to an ideal in $\boldsymbol{A}$). It is well known (see [1, Theorem 3.32], for example) that any such submodule has a finite basis which generates it in the sense that each of its elements can be expressed as a sum of (polynomial) multiples of the basis elements. We shall show that each basis of $\boldsymbol{M}$ can be reduced to one containing precisely two elements. In general, many essentially distinct two-element bases exist; this is in contrast to the situation in $\boldsymbol{A}$ where each ideal has a uniquely defined one-element basis, namely, the monic generator of minimal degree. However, one basis of $\boldsymbol{M}$ is easily available, namely, $\mathcal{B} = \{(g, 1), (x^n, 0)\}$. The central theme of our method is *to use the conditions* (3) *to define a*

basis $\mathcal{B}'$ that contains the required solution. Our algorithms represent different ways of deriving $\mathcal{B}'$ from $\mathcal{B}$.

We shall need a classification of Gröbner bases of submodules of $\boldsymbol{A}^2$: since there is only one variable involved this is very elementary and follows straightforwardly from the general theory (see [1]). However, in order to make this paper self-contained, we have devoted Section II to developing the necessary theory from first principles. We define for each integer r a term order $<_r$ in $\boldsymbol{A}^2$. When $r = \ell - m$ the required solution becomes the *minimal element* in $\boldsymbol{M}$, uniquely defined up to multiplication by scalars (elements of $\boldsymbol{F}$). This element must appear in any Gröbner basis of $\boldsymbol{M}$ under $<_r$ and, consequently, it can be determined by finding such a Gröbner basis. The solution of (1) arises when $r = -1$, the solution of Berlekamp's equation defined above arises when $r = 0$, while, in the errors-and-erasures problem, r is the number of erasures. A flow diagram is given for the $r \geq 0$ case which can be directly compared to BM.

We note in passing that Gröbner basis techniques have also been applied to the decoding problem (in a different way) in [8]–[10].

In Section III we prove that $\mathcal{B}$ is a Gröbner basis relative to the term order $<_{\delta g}$ and present algorithms corresponding to PGZ and E for deriving a Gröbner basis $\mathcal{B}'$ relative to $<_r$, for any r. Both of these algorithms use the full expansion of g at the outset. In contrast, Section IV describes an "iterative" algorithm similar to BM, in which successive partial expansions of g are used. All our algorithms hold equally well with $\boldsymbol{F}$ replaced by any field in which exact computation is possible, so for simplicity and to enable the reader to follow the calculations easily by hand, most of the examples are over the field $\boldsymbol{Q}$ of rational numbers.

The main motivation for this paper is the derivation of algorithms for solving the general key equation (2) from a unified and conceptually simple theoretical standpoint, namely, that of choosing a minimal element in $\boldsymbol{M}$ defined by the conditions (3). Once the basic ideas of Gröbner bases have been assimilated, the constructions are all quite natural. Not surprisingly, perhaps, this approach leads to algorithms which, when specialized to the appropriate values of r, are computationally at least as good as their classical counterparts. It follows that a decoder designed to implement one of our algorithms will be at least as good as, and in the case of that corresponding to BM, better than its classical counterpart. Consequently, we claim that it is no longer true that, "when building a decoder [for alternant codes], one gives up the conceptually clear in favor of the computationally efficient" [3, p. 183].

Finally, we remark that our techniques have also been applied to give a new derivation of the Welch–Berlekamp algorithm [24]. Moreover, they can be generalized to key equations in several variables, and so can be applied (at least in principle) to decoding certain geometric Goppa codes [17].

II. Gröbner Bases in A^2

Since we are working with polynomials in only one variable, it is easy to describe the possibilities for Gröbner bases of submodules of A^2. Moreover, given a basis of a submodule and a term order it is immediate that the construction of the relevant Gröbner basis is only a matter of *reduction*, in other words, there is no need to generate *critical pairs*. Thus Buchberger's algorithm [6] is not required and, as a consequence, the theory is very elementary and easily developed from first principles: this is our aim in this section. The reader is referred to [1] for a comprehensive treatment of the theory of Gröbner bases, while [11] gives an alternative introduction from a different perspective.

Each element of A^2 can be written as a linear combination of *terms*, that is, of elements of the form

$$\{(x^i, 0), (0, x^j): i, j \geq 0\}.$$

Thus

$$\begin{aligned}(3x^2 + x - 1, x^3 + 2) =& 3(x^2, 0) + (x, 0)\\ &- (1, 0) + (0, x^3) + 2(0, 1)\end{aligned}$$

for example. (Note that, following [1], our terms do not have coefficients attached.) The problem one immediately encounters, just as with polynomials in several variables, is that there is no uniquely defined order on the terms corresponding to the degree ordering on the terms (of the form x^k) in $\boldsymbol{A}$. The essential properties that a *term order* must satisfy are that every pair of terms be comparable, that there be no infinitely descending chains of terms, and that the term order be compatible with multiplication by x^k, that is, if $\boldsymbol{t}_1$, $\boldsymbol{t}_2$ are terms with $\boldsymbol{t}_1 < \boldsymbol{t}_2$ then $x^k\boldsymbol{t}_1 < x^k\boldsymbol{t}_2$. For our purposes, it will be sufficient to define such a term order $<_r$ with respect to an integer parameter r by the following conditions:

1) $(x^i, 0) <_r (x^{i'}, 0)$ for $i < i'$ and $(0, x^j) <_r (0, x^{j'})$ for $j < j'$;
2) $(x^i, 0) <_r (0, x^j)$ if and only if $i \leq j + r$.

It is easy to see that the required conditions are satisfied under this definition. The term orders defined in this way can be explicitly given as follows:

1) If $r \geq 0$ then $(1, 0) <_r (x, 0) <_r \cdots (x^r, 0) <_r (0, 1) <_r (x^{r+1}, 0) <_r (0, x) <_r \cdots$
2) If $r < 0$ then $(0, 1) <_r (0, x) <_r \cdots (0, x^{-r-1}) <_r (1, 0) <_r (0, x^{-r}) <_r (x, 0) <_r \cdots$.

For the remainder of this section we fix a specific term order $<_r$, briefly $<$, according to this definition. Each $(a, b) \in \boldsymbol{A}^2$ can be expressed as a linear combination of terms under $<$; that which is greatest is defined to be the *leading term* of (a, b) and denoted $\mathrm{Lt}(a, b)$. The coefficient of the leading term is called the *leading coefficient* and denoted $\mathrm{Lc}(a, b)$. For any nonempty subset $\mathcal{S} \subseteq \boldsymbol{A}^2$ we denote by $\mathrm{Lt}(\mathcal{S})$ the set of leading terms of elements of $\mathcal{S}$. The symbol $\langle\mathcal{S}\rangle$ denotes the submodule of $\boldsymbol{A}^2$ generated by $\mathcal{S}$, that is, its closure under addition and multiplication by polynomials; in particular, $\langle\mathrm{Lt}(\mathcal{S})\rangle$ is the submodule generated by the leading terms of the elements of $\mathcal{S}$.

The term $\boldsymbol{t}_1$ is said to be *divisible* by the term $\boldsymbol{t}_2$ if there exists an x^k such that $\boldsymbol{t}_1 = x^k\boldsymbol{t}_2$. If

$$\mathcal{S} = \{(a_i, b_i): 1 \leq i \leq m\} \subseteq \boldsymbol{A}^2$$

we can divide an arbitrary element (a, b) by $\mathcal{S}$ using the following straightforward generalization of the division algorithm in $\boldsymbol{A}$. A set of polynomials ("quotients") f_i and a remainder (u, v) are determined so that

$$(a, b) = f_1(a_1, b_1) + f_2(a_2, b_2) + \cdots + f_m(a_m, b_m) + (u, v)$$

where either no term of (u, v) is divisible by any $\mathrm{Lt}(a_i, b_i)$ or else $(u, v) = (0, 0)$. Our presentation is a straightforward analog of [11, Theorem 2.3.3] so we give an outline proof only.

Algorithm 2.1:

Input:

$(a_1, b_1), \cdots, (a_m, b_m), (a, b)$, a term order $<$

Output:

$f_1, \cdots, f_m \in \boldsymbol{A}$, $(u, v) \in \boldsymbol{A}^2$ such that

$$(a, b) = \sum_{i=1}^{m} f_i(a_i, b_i) + (u, v)$$

where either no term of (u, v) is divisible by any of

$$\mathrm{Lt}(a_i, b_i),\ 1 \le i \le m$$

or $(u, v) = (0, 0)$.

Initialize:

$f_1 := 0; \cdots; f_m := 0 \in \boldsymbol{A}$; $(u, v) := (0, 0)$; $(c, d) := (a, b)$

WHILE $(c, d) \neq (0, 0)$ DO

$i := 1$

division_occurred := false

WHILE $i \le m$ AND division_occurred=false DO

IF $\mathrm{Lt}(c, d)$ is a multiple of $\mathrm{Lt}(a_i, b_i)$ THEN

$$f_i := f_i + \frac{\mathrm{Lc}(c, d)\mathrm{Lt}(c, d)}{\mathrm{Lc}(a_i, b_i)\mathrm{Lt}(a_i, b_i)}$$

$$(c, d) := (c, d) - \frac{\mathrm{Lc}(c, d)\mathrm{Lt}(c, d)}{\mathrm{Lc}(a_i, b_i)\mathrm{Lt}(a_i, b_i)}(a_i, b_i)$$

division_occurred := true

ELSE

$i := i + 1$

IF division_occurred = false THEN

$(u, v) := (u, v) + \mathrm{Lc}(c, d)\mathrm{Lt}(c, d)$

$(c, d) := (c, d) - \mathrm{Lc}(c, d)\mathrm{Lt}(c, d)$

Proof: At each stage the algorithm compares $\mathrm{Lt}(c, d)$ with the leading terms of the elements of $\mathcal{S}$. If it is divisible by some $\mathrm{Lt}(a_i, b_i)$ then an appropriate multiple of (a_i, b_i) is subtracted from (c, d) to cancel its leading term and the multiplier is added to the quotient f_i. If it is not divisible by any $\mathrm{Lt}(a_i, b_i)$ then the leading term, with its coefficient attached, is moved to the remainder (u, v). The algorithm halts when (c, d) has been reduced to $(0, 0)$. It is clear that this occurs after finitely many steps because at each pass through the main loop $\mathrm{Lt}(c, d)$ is replaced by a term that comes before it under the given term order. It is also clear that (u, v) has the required property. □

Now let $\boldsymbol{M}$ be an arbitrary nonzero submodule of $\boldsymbol{A}^2$. Among those terms of the form $(x^i, 0) \in \mathrm{Lt}(\boldsymbol{M})$, if any such exist, there is a smallest value of i that occurs, say $i = p$; it is obvious that every other term in $\mathrm{Lt}(\boldsymbol{M})$ of the form $(x^i, 0)$ is a multiple of $(x^p, 0)$ and so can be omitted from any generating set for $\langle \mathrm{Lt}(\boldsymbol{M}) \rangle$. A similar statement holds for terms of the form $(0, x^j)$, if any such exist, and we define $j = q$ to be the smallest exponent occurring. Throughout the rest of the paper the symbols p, q will be reserved for these minimal exponents.

Next consider the case in which there is no term of the form $(0, x^j)$ in $\mathrm{Lt}(\boldsymbol{M})$; then

$$\langle \mathrm{Lt}(\boldsymbol{M}) \rangle = \langle (x^p, 0) \rangle = \{ f(x^p, 0) : f \in \boldsymbol{A} \}.$$

Let $(a_1, b_1) \in \boldsymbol{M}$ have leading term $(x^p, 0)$. Then, for any $(a, b) \in \boldsymbol{M}$, the division algorithm gives

$$(a, b) = f(a_1, b_1) + (u, v)$$

with no term of (u, v) divisible by $(x^p, 0)$, or with $(u, v) = (0, 0)$. If $(u, v) \neq (0, 0)$ then, since

$$(u, v) = (a, b) - f(a_1, b_1) \in \boldsymbol{M}$$

$\mathrm{Lt}(u, v)$ has the form $(x^i, 0)$ but is not divisible by $(x^p, 0)$ contrary to the definition of p. It follows that $(u, v) = (0, 0)$ and $\boldsymbol{M} = \langle (a, b) \rangle$, and indeed, that (a, b) is the unique generator of $\boldsymbol{M}$ up to a scalar multiple. A similar argument holds if there is no element in $\mathrm{Lt}(\boldsymbol{M})$ of the form $(x^i, 0)$. We shall not be concerned with either of these cases in the ensuing discussion so from now on we consider only modules $\boldsymbol{M}$ satisfying the following.

Basis Assumption:

$\langle \mathrm{Lt}(\boldsymbol{M}) \rangle$ has a basis of the form $\{(x^p, 0), (0, x^q)\}$.

Let

$$\mathcal{B} = \{(a_1, b_1), (a_2, b_2)\} \subseteq \boldsymbol{M}$$

where

$$\mathrm{Lt}(a_1, b_1) = (x^p, 0),\ \mathrm{Lt}(a_2, b_2) = (0, x^q).$$

Then $\langle \mathrm{Lt}(\mathcal{B}) \rangle = \langle \mathrm{Lt}(\boldsymbol{M}) \rangle$, and for any $(a, b) \in \boldsymbol{M}$, the division algorithm gives

$$(a, b) = f(a_1, b_1) + g(a_2, b_2) + (u, v)$$

with no term of (u, v) divisible by $(x^p, 0)$ or $(0, x^q)$, or with $(u, v) = (0, 0)$. Arguing as above from the definition of p, q, we have $(u, v) = (0, 0)$ and $\boldsymbol{M} = \langle (a_1, b_1), (a_2, b_2) \rangle$. Thus $\mathcal{B}$ is a basis of $\boldsymbol{M}$. Moreover, any subset $\mathcal{S} \subseteq \boldsymbol{M}$ containing $\mathcal{B}$ is also a basis and satisfies $\langle \mathrm{Lt}(\mathcal{S}) \rangle = \langle \mathrm{Lt}(\boldsymbol{M}) \rangle$. By definition, any basis $\mathcal{S}$ of $\boldsymbol{M}$ satisfying $\langle \mathrm{Lt}(\mathcal{S}) \rangle = \langle \mathrm{Lt}(\boldsymbol{M}) \rangle$ is a *Gröbner basis*. In other words, $\mathcal{S}$ is a Gröbner basis if the leading term of each element of $\boldsymbol{M}$ is divisible by the leading term of some element of $\mathcal{S}$. Obviously, every Gröbner basis must contain a subset of the form $\mathcal{B}$ and it is also clear that if the set

$$\{(a_1, b_1), (a_2, b_2)\} \subseteq \boldsymbol{M}$$

with

$$\mathrm{Lt}(a_1, b_1) = (x^i, 0) \qquad \mathrm{Lt}(a_2, b_2) = (0, x^j)$$

is a basis of $\boldsymbol{M}$ then $i = p$, $j = q$ and the set is a Gröbner basis.

In general, the remainder (u, v) produced by the division algorithm is not uniquely defined by (a, b) and the set $\mathcal{S}$; it depends on the order of the elements of $\mathcal{S}$. For example, if

$$\mathcal{S} = \{(x^2 + 1, x), (x + 1, 1)\}$$

then

$$(x^2 + x + 1, x - 1) = (x^2 + 1, x) + (x + 1, 1) + (-1, -2)$$

whereas

$$(x^2 + x + 1, x - 1) = x(x + 1, 1) + (1, -1).$$

However, let

$$\mathcal{S} = \{(a_k, b_k): 1 \leq k \leq m\}$$

be a Gröbner basis of $\boldsymbol{M}$; then division of (a, b) by $\mathcal{S}$ gives a uniquely defined remainder. This follows because if

$$(a, b) = \sum_{k=1}^{m} f_k(a_k, b_k) + (u, v)$$
$$(a, b) = \sum_{k=1}^{m} f'_k(a_k, b_k) + (u', v')$$

where (u, v), (u', v') each satisfy the property required of a remainder, then by subtraction

$$(u, v) - (u', v') = \sum_{k=1}^{m} (f'_k - f_k)(a_k, b_k)$$

lies in $\boldsymbol{M}$ and by considering its leading term we deduce that it must be $(0, 0)$.

The unique remainder on division of (a, b) by a Gröbner basis $\mathcal{S}$ is called its *normal form* (with respect to $\mathcal{S}$); we denote it by $\mathrm{Nf}_{\mathcal{S}}(a, b)$, or simply $\mathrm{Nf}(a, b)$ if no confusion would arise. Thus $(a, b) \in \boldsymbol{M}$ if and only if $\mathrm{Nf}_{\mathcal{S}}(a, b) = (0, 0)$ (and this is true for *any* Gröbner basis $\mathcal{S}$ of $\boldsymbol{M}$). If every element (a_k, b_k) in the Gröbner basis $\mathcal{S}$ has the property that none of its terms is divisible by the leading term of any (a_i, b_i), for $i \neq k$, then $\mathcal{S}$ is called a *reduced* Gröbner basis. It is obvious that a reduced Gröbner basis must consist of precisely two elements whose components satisfy certain degree conditions.

For convenient reference we summarize the foregoing remarks in the following lemma.

Lemma 2.2: Let $<_r$ be a term order on $\boldsymbol{A}^2$ and let $\boldsymbol{M}$ be a submodule of $\boldsymbol{A}^2$ with

$$\langle \mathrm{Lt}(\boldsymbol{M}) \rangle = \langle (x^p, 0), (0, x^q) \rangle.$$

a) A subset $\mathcal{S} \subseteq \boldsymbol{M}$ is a Gröbner basis of $\boldsymbol{M}$ if and only if it contains a subset

$$\mathcal{B} = \{(a_1, b_1), (a_2, b_2)\}$$

with $\mathrm{Lt}(a_1, b_1) = (x^p, 0)$, $\mathrm{Lt}(a_2, b_2) = (0, x^q)$.

b) If

$$\mathcal{C} = \{(a_1, b_1), (a_2, b_2)\}$$

is a basis of $\boldsymbol{M}$ and

$$\mathrm{Lt}(a_1, b_1) = (x^i, 0) \qquad \mathrm{Lt}(a_2, b_2) = (0, x^j)$$

then $i = p$, $j = q$, and $\mathcal{C}$ is a Gröbner basis.

c) A reduced Gröbner basis of $\boldsymbol{M}$ has the form

$$\{(a_1, b_1), (a_2, b_2)\}$$

where

$$\mathrm{Lt}(a_1, b_1) = (x^p, 0) \qquad \mathrm{Lt}(a_2, b_2) = (0, x^q)$$

and the following hold:

i) $p > \delta(b_1) + r$ and $\delta(a_2) \leq q + r$

ii) $\delta(a_2) < p$ and $\delta(b_1) < q$. □

If $\mathrm{Lt}(a, b)$ has the form $(x^i, 0)$ (resp., $(0, x^j)$), it is convenient to say that the leading term of (a, b) is "on the left" (resp., "on the right"). We refer to the property defined in part b) of the lemma by saying that $\mathcal{C}$ consists of two elements with leading terms "on opposite sides." Thus parts a) and b) give the following alternative characterizations of a Gröbner basis of $\boldsymbol{M}$.

Corollary 2.3: The following define Gröbner bases of $\boldsymbol{M}$.

i) $\mathcal{S} \subseteq \boldsymbol{M}$ is a subset containing a pair of elements with leading terms on opposite sides and exponents minimal;

ii) $\mathcal{C} \subseteq \boldsymbol{M}$ is a two-element basis whose elements have leading terms on opposite sides. □

We shall use these conditions frequently throughout the paper.

The ordering of the terms allows us to define a *minimal element* of $\boldsymbol{M}$ as one whose leading term is minimal among those in $\mathrm{Lt}(\boldsymbol{M})$. Let (a_1, b_1), (a_2, b_2) be defined as in part a) of the lemma. It is clear that one or other of these elements is a minimal element: if $p \leq q + r$ then $\mathrm{Lt}(a_1, b_1) < \mathrm{Lt}(a_2, b_2)$ and (a_1, b_1) is minimal, while if $p > q + r$ then $\mathrm{Lt}(a_1, b_1) > \mathrm{Lt}(a_2, b_2)$ and (a_2, b_2) is minimal. The essential properties of a minimal element are given in the following lemma.

Lemma 2.4:

a) A minimal element of $\boldsymbol{M}$ is unique up to scalar multiples.

b) Every Gröbner basis of $\boldsymbol{M}$ contains a minimal element.

Proof: If (a, b), (a', b') are both minimal then the leading term of the remainder on division of (a, b) by (a', b') cannot be less than $\mathrm{Lt}(a', b')$ and so the remainder is $(0, 0)$. Thus (a', b') divides (a, b) and similarly (a, b) divides (a', b') and we conclude that the quotients are constants. Part b) follows directly from part a) of the previous lemma (or, indeed, from the definition of a Gröbner basis). □

From now on we refer to *the* minimal element of $\boldsymbol{M}$ with understanding that this is defined up to a constant multiple. As a general principle we derive the minimal element (a, b) normalized (if necessary) to have leading coefficient 1. In some situations (such as the decoding application) normalizations with b monic or having constant coefficient 1 may be more appropriate: these will be used in the examples.

Starting with a given (two-element) Gröbner basis $\mathcal{B}$ of $\boldsymbol{M}$ relative to a certain term order $<_r$, our aim is to determine

another basis $\mathcal{B}'$, which is a Gröbner basis relative to another term order $<_{r'}$. We shall present three ways in which $\mathcal{B}'$ can be constructed. The first two are not restricted to the particular problem under consideration (namely, solving (2)), so we end this section by describing them informally in general.

Method 1 (Change of Term Order [14], [17]): Let $\mathrm{Nf}(\boldsymbol{t})$ denote the normal form of the term $\boldsymbol{t}$ relative to $\mathcal{B}$. Let the terms $\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots \in \boldsymbol{A}^2$ be ordered according to $<_{r'}$, and consider the sequence $\{\mathrm{Nf}(\boldsymbol{t}_1), \mathrm{Nf}(\boldsymbol{t}_2), \cdots\}$. If

$$\mathrm{Nf}(\boldsymbol{t}_j) = \sum_{i<j} \alpha_i \mathrm{Nf}(\boldsymbol{t}_i) \tag{4}$$

where $\alpha_i \in \boldsymbol{F}$ then

$$(a_1, b_1) = \boldsymbol{t}_j - \sum_{i<j} \alpha_i \boldsymbol{t}_i$$

lies in $\boldsymbol{M}$ because its normal form is $(0, 0)$. Considering the normal forms in turn, let j be the first index for which an equation of this form holds. Thus $\boldsymbol{t}_j$ is the least leading term, relative to $<_{r'}$, of any element in $\boldsymbol{M}$. Omit from further consideration all subsequent terms which are multiples of $\boldsymbol{t}_j$ and continue to consider the normal forms of the $\boldsymbol{t}_i$ relative to $<_{r'}$. (The set of terms still to be considered is nonempty by the Basis Assumption.) If k is the next index for which an equation of the form (4) holds then, in an obvious notation

$$(a_2, b_2) = \boldsymbol{t}_k - \Sigma_{i<k} \beta_i \boldsymbol{t}_i \in \boldsymbol{M}.$$

Set

$$\mathcal{B}' = \{(a_1, b_1), (a_2, b_2)\}.$$

Since the leading terms of the two elements of $\mathcal{B}'$ (relative to $<_{r'}$) are on opposite sides with exponents minimal, it is the required Gröbner basis by Corollary 2.3 i). □

Method 2 (Analog of the Euclidean Algorithm): $\mathcal{B}$ is a basis of $\boldsymbol{M}$, supposed not to be a Gröbner basis relative to $<_{r'}$ (otherwise there is nothing to do), so using Lemma 2.2 b), we find that the leading terms relative to $<_{r'}$ of the two elements $(a_1, b_1), (a_2, b_2)$ of $\mathcal{B}$ are either both on the left or both on the right. Thus one of the leading terms is a multiple of the other and so one basis element can be divided by the other using the division algorithm. If, without loss of generality

$$(a_1, b_1) = f(a_2, b_2) + (u, v)$$

then we can replace $\mathcal{B}$ by the new basis $\{(a_2, b_2), (u, v)\}$, where, in particular, $\mathrm{Lt}(u, v) <_{r'} \mathrm{Lt}(a_2, b_2)$. This process may be repeated until no further division is possible; it is finite because the sequence of leading terms of the remainders is decreasing, and it ends with a basis having leading terms on opposite sides which is the required Gröbner basis by Corollary 2.3 ii). □

III. The Solution Module Analogs of PGZ and E

The multivariable theoretical foundation for this section is contained in [18]. Let $\boldsymbol{M}$ be the submodule of $\boldsymbol{A}^2$ comprising all the solutions of (2). The first result gives a natural basis of $\boldsymbol{M}$.

Lemma 3.1:

$$\mathcal{B} = \{(g, 1), (x^n, 0)\}$$

is a reduced Gröbner basis of $\boldsymbol{M}$ relative to the term order $<_{\delta g}$.

Proof: If $(a, b) \in \boldsymbol{M}$ then $a - bg$ is a multiple of x^n so

$$(a, b) = b(g, 1) + (a - bg, 0) = b(g, 1) + f(x^n, 0) \in \langle \mathcal{B} \rangle$$

for some $f \in \boldsymbol{F}[x]$. Since both elements of $\mathcal{B}$ are clearly in $\boldsymbol{M}$ we deduce that $\mathcal{B}$ is a basis. Now $\mathrm{Lt}(g, 1) = (0, 1)$ and $\mathrm{Lt}(x^n, 0) = (x^n, 0)$ with respect to the given term order, so it follows from Corollary 2.3 ii) that $\mathcal{B}$ is a Gröbner basis; it is reduced because no term of $(g, 1)$ is divisible by $(x^n, 0)$ (recall that by assumption $\delta(g) \leq n - 1$) and $(x^n, 0)$ is not divisible by $(0, 1)$. □

Our aim is to force the required solution defined by (3) to be the minimal element of the solution module relative to a new term order so that it will necessarily be contained in any Gröbner basis relative to that order. To this end we have the following result.

Theorem 3.2: The solution (a, b) of (2) defined by the conditions (3) is the minimal element of $\boldsymbol{M}$ relative to the term order $<_r$ where $r = \ell - m$.

Proof: We consider only the case in which $\mathrm{Lt}(a, b)$ is on the left, the argument for the other alternative being a straightforward analog. Thus suppose that $\mathrm{Lt}(a, b) = (x^{\delta a}, 0)$ and let $(a', b') \in \boldsymbol{M}$ satisfy $\mathrm{Lt}(a', b') <_r \mathrm{Lt}(a, b)$. The congruences $a \equiv bg \bmod x^n$ and $a' \equiv b'g \bmod x^n$ imply $ab' \equiv a'b \bmod x^n$. Now if $\mathrm{Lt}(a', b') = (x^{\delta a'}, 0)$ then $\delta a' < \delta a \leq \ell$ while

$$\delta b' < \delta a' - r < \delta a - r \leq \ell - (\ell - m) = m.$$

Similarly, if $\mathrm{Lt}(a', b') = (0, x^{\delta b'})$ then $\delta a' \leq \delta b' + r < \delta a \leq \ell$ while

$$\delta b' < \delta a - r \leq \ell - (\ell - m) = m.$$

In both cases it follows that

$$\delta(ab') < n, \delta(a'b) < n$$

which implies that $ab' = a'b$. But a and b are relatively prime so a divides a' which is impossible since $\delta a' < \delta a$. This completes the proof. □

Recall that the required solution (ω, σ) of (1) is defined by the properties $\delta\omega < \delta\sigma \leq t$, and ω, σ relatively prime. The preceding theorem shows that the condition on the degrees may be replaced by the weaker conditions $\delta\omega \leq t-1$, $\delta\sigma \leq t$. This gives the first part of the following result which should be compared with [28, Theorem 8.5]. For Berlekamp's equation the conditions are $\delta\hat{\omega} \leq t$, $\delta\sigma \leq t$, while for the errors-and-erasures equation we have $\delta\lambda \leq v + e$, $\delta\mu \leq v$ with v the number of errors, e the number of erasures, and $2v + e \leq 2t$.

Corollary 3.3: With the appropriate definition of solution module

i) The required solution (ω, σ) of (1) is the minimal element relative to $<_{-1}$.
ii) The required solution $(\hat{\omega}, \sigma)$ of Berlekamp's equation is the minimal element relative to $<_0$.
iii) The required solution (λ, μ) of the errors-and-erasures equation is the minimal element relative to $<_e$. □

From the results of the previous section we have

Corollary 3.4: The required solution of (2) is contained in any Gröbner basis relative to $<_r$ where $r = \ell - m$. □

In the remainder of this section we consider the implementation of the two techniques described at the end of Section II and give some examples of applications to the solution of (1).

For $k < n$, the terms $(x^k, 0)$ are in normal form relative to the basis $\mathcal{B} = \{(g, 1), (x^n, 0)\}$ defined in Lemma 3.1, while for $(0, x^k)$ we have

$$\begin{aligned}\mathrm{Nf}(0, x^k) &= x^k(g, 1) - (x^k g, 0)\\ &= x^k(g, 1) - f(x^n, 0) - (\overline{x^k g}, 0)\end{aligned}$$

for some f, where the bar denotes reduction modulo x^n. Note that Nf is defined relative to $\mathcal{B}$ (or an appropriate specific example) throughout the rest of this section. Thus

$$\mathrm{Nf}(0, x^k) = (-\overline{x^k g}, 0).$$

Method 1, described at the end of the previous section, gives rise to the following algorithm which determines the minimal element as the first to be inserted into the new basis.

Algorithm 3.5:

Input:
g, n, ℓ, m; terms $\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots$ ordered by $<_r$ with $r = \ell - m$
Output:
$(a, b) \in \boldsymbol{M}$ satisfying (3)
Initialize:
$j := 1$; not_done := true
WHILE not_done DO
  $j := j + 1$
  IF there exist α_i with
  $\mathrm{Nf}(\boldsymbol{t}_j) = \sum_{i<j} \alpha_i \mathrm{Nf}(\boldsymbol{t}_i)$ THEN
    $(a, b) := \boldsymbol{t}_j - \sum_{i<j} \alpha_i \mathrm{Nf}(\boldsymbol{t}_i)$
    not_done := false

Next we demonstrate that in the case $r = -1$ the solution of (1) by this algorithm is computationally equivalent to PGZ . If

$$h = h_{2t-1}x^{2t-1} + \cdots + h_0$$

then the normal forms required are

$$\begin{aligned}\mathrm{Nf}(0, 1) &= (-h_{2t-1}x^{2t-1} - \cdots - h_0, 0)\\ \mathrm{Nf}(1, 0) &= (1, 0)\\ \mathrm{Nf}(0, x) &= (-h_{2t-2}x^{2t-1} - \cdots - h_0 x, 0)\\ \mathrm{Nf}(x, 0) &= (x, 0)\\ &\cdots\\ \mathrm{Nf}(x^{t-1}, 0) &= (x^{t-1}, 0)\\ \mathrm{Nf}(0, x^t) &= (-h_{t-1}x^{2t-1} - \cdots - h_0 x^t, 0).\end{aligned}$$

We need to determine the first normal form in this list that is a linear combination of those that precede it. To see what is involved here we consider the following example.

TABLE I
EXAMPLE 3.6

	x^7	x^6	x^5	x^4	x^3	x^2	x	1
Nf(0, 1)	0	1	−2	2	−1	1	0	−1
Nf(1, 0)								1
Nf(0, x)	1	−2	2	−1	1	0	−1	
Nf(x, 0)							1	
Nf(0, x^2)	−2	2	−1	1	0	−1		
Nf(x^2, 0)						1		
Nf(0, x^3)	2	−1	1	0	−1			
Nf(x^3, 0)					1			
Nf(0, x^4)	−1	1	0	−1				

Example 3.6: The initial terms of a linear recurring sequence over $\boldsymbol{Q}$ are 1, 0, −1, 1, −2, 2, −1, 0. We seek the minimal polynomial of the sequence given that it has degree at most 4. This problem is precisely that defined by (1) with the required minimal polynomial being determined as the "reciprocal" of σ (i.e., $x^{-\delta(\sigma)}\sigma(1/x)$), and with

$$h = -x^6 + 2x^5 - 2x^4 + x^3 - x^2 + 1, \quad x^{2t} = x^8.$$

The calculations proceed as in Table I. Since the second component is always 0, only the first components of the normal forms are shown; all omitted entries are 0.

Observe that the normal forms $\mathrm{Nf}(x^i, 0)$ can be used to reduce successively the number of columns of this array that need to be considered. As soon as $\mathrm{Nf}(0, x^4)$ has been included there must be a solution since at that stage we are essentially seeking a linear relation among the rows of a 5×4 matrix. We find that

$$\begin{aligned}-\mathrm{Nf}(0, 1) + \mathrm{Nf}(1, 0) - \mathrm{Nf}(0, x) + \mathrm{Nf}(x, 0) - \mathrm{Nf}(x^2, 0)\\ = \mathrm{Nf}(0, x^4)\end{aligned}$$

and hence that

$$(\omega, \sigma) = (-x^2 + x + 1, x^4 + x + 1).$$

It follows that the minimal polynomial is $x^4 + x^3 + 1$. □

In general, let H_k, $0 \le k \le t - 1$ be the matrix

$$H_k = \begin{pmatrix} -h_{2t-1} & -h_{2t-2} & \cdots & -h_k \\ -h_{2t-2} & -h_{2t-3} & \cdots & -h_{k-1} \\ & \cdots & & \\ -h_{2t-1-k} & -h_{2t-2-k} & \cdots & -h_0 \end{pmatrix}.$$

The minimal element in the new basis appears at the first k for which the rank of H_k equals the rank of H_{k-1} and it is clear that $k \le t$. If, for this value of k, we define

$$Y = (-h_{2t-1-k}, -h_{2t-2-k}, \cdots, -h_0)$$

and if $X = (X_0, \cdots, X_{k-1})$ is the solution of the system $XH = Y$, then the minimal element is

$$\left(-\sum_{i=0}^{k-1}\sum_{j=0}^{i} X_j h_{i-j} x^i, \quad x^k - \sum_{i=0}^{k-1} X_i x^i\right).$$

TABLE II
EXAMPLE 3.9

(a_2, b_2)	(a_1, b_1)	f
$(x^8, 0)$	$(-x^6+2x^5-2x^4+x^3-x^2+1, 1)$	$-x^2-2x-2$
$(-x^6+2x^5-2x^4+x^3-x^2+1, 1)$	$(x^5-3x^4-x^2+2x+2, x^2+2x+2)$	$-x-1$
$(x^5-3x^4-x^2+2x+2, x^2+2x+2)$	$(-5x^4+4x+3, x^3+3x^2+4x+3)$	$-\frac{1}{5}x+\frac{3}{5}$
$(-5x^4+4x+3, x^3+3x^2+4x+3)$	$(-\frac{1}{5}x^2+\frac{1}{5}x+\frac{1}{5}, \frac{1}{5}x^4+\frac{1}{5}x+\frac{1}{5})$	—

Algorithm PGZ can be described as finding the maximum value of s such that the top left $s \times s$ submatrix L of H_t is invertible and then solving an appropriate system of equations using L as coefficient matrix. This is precisely the same as the foregoing computation. In fact, we use the syndromes (coefficients of h) in the opposite order to that in [33, Theorem 9.9], but nevertheless the factorization of L as a product of two Vandermonde matrices and a diagonal matrix remains valid and equivalent calculations may be carried out to determine the error locations and values.

Method 2 of the previous section leads to the following computation. We begin with the basis elements

$$(x^n, 0)$$
$$(g_{n'}x^{n'} + \cdots + g_0, 1)$$

where $n' < n$ is the largest index of a nonzero coefficient of g. If the leading terms of these elements are on opposite sides (i.e., if $n' \le 0 + r = r$ so that the leading term of the second element is on the right) then they already form a Gröbner basis relative to $<_r$ and the minimal element is $(g, 1)$, since

$$n > \ell + m \ge |\ell - m| = |r| \ge r.$$

If the leading terms are on the same side (the left) then we may calculate

$$(x^n, 0) = \frac{1}{g_{n'}} x^{n-n'}(g_{n'}x^{n'} + \cdots + g_0, 1) + \frac{1}{g_{n'}}(-g_{n'-1}x^{n-1} - \cdots - g_0 x^{n-n'}, -x^{n-n'})$$

and hence

$$\left\{(g, 1), \frac{1}{g_{n'}}(-g_{n'-1}x^{n'-1} - \cdots - g_0 x^{n-n'}, -x^{n-n'})\right.$$

is a new basis. This process is iterated until the first element appears with leading term on the right. At that stage, the basis is a Gröbner basis relative to $<_r$ (by Corollary 2.3 ii). Since the element with leading term on the right was the last generated, it has the smaller leading term and so is the minimal element. More formally, we have the following algorithm in which the comparison with Algorithm E is apparent (cf. [27], [28]).

Algorithm 3.7:

Input: g, n, ℓ, m;
Output: $(a, b) \in \boldsymbol{M}$ satisfying (3)
Initialize:
$(a_1, b_1) := (x^n, 0)$; $(a_2, b_2) := (g, 1)$; $r := \ell - m$
WHILE $\delta a_2 > \delta b_2 + r$ DO
  $(u, v) := (a_1, b_1) \bmod \{(a_2, b_2)\}$
    [using the division algorithm]
  $(a_1, b_1) := (a_2, b_2)$
  $(a_2, b_2) := (u, v)$
$(a, b) := \frac{1}{\mathrm{Lc}\,(a_2, b_2)}(a_2, b_2)$

□

Remark 3.8: We note that in the decoding application (1) the stopping criterion here (namely, stop when a Gröbner basis is achieved) is more natural than that used in the technique based on the Euclidean algorithm (namely, stop when the degree of the remainder drops below t): our algorithm actually halts, whereas the Euclidean algorithm runs on to obtain the greatest common divisor of h and x^{2t}.

Example 3.9: We treat the same example as above: the computations are summarized in Table II with the f appearing in the division algorithm recorded for convenience. After normalization we obtain the same minimal element as before. □

IV. SOLUTION BY APPROXIMATIONS ANALOG OF BM

In this section we derive a technique corresponding to BM.

Method 3 (Solution by Approximations): The idea is to solve the sequence of partial problems

$$a \equiv bg \bmod x^k, \quad 0 \le k \le n$$

determining Gröbner bases relative to $<_r$ for the solution modules $\boldsymbol{M}_0, \boldsymbol{M}_1, \cdots, \boldsymbol{M}_n = \boldsymbol{M}$.

A. The General Algorithm

If $(a, b) \in \boldsymbol{M}_k$ then we wish to determine whether or not $(a, b) \in \boldsymbol{M}_{k+1}$ and if not to use it to obtain an element of $\boldsymbol{M}_{k+1}$. First, observe that $(a, b) \notin \boldsymbol{M}_{k+1}$ precisely when the coefficient of x^k in the expansion of $bg - a$ is nonzero. More explicitly, let Nf denote normal form relative to the Gröbner basis $\{(\overline{g}, 1), (x^{k+1}, 0)\}$ of $\boldsymbol{M}_{k+1}$, where the bar denotes reduction modulo x^{k+1}, and let

$$b\overline{g} - a \equiv \alpha x^k \bmod x^{k+1}$$

so that

$$b\bar{g} - a = \alpha x^k + f x^{k+1}$$

for some f. Note that this implies

$$bg - a = \alpha x^k + f' x^{k+1}$$

for some f', and hence that α is also the coefficient of x^k in the expansion of $bg - a$. Now

$$\begin{aligned}\mathrm{Nf}(a, b) &= \mathrm{Nf}[b(\bar{g}, 1) - (b\bar{g} - a, 0)] \\ &= \mathrm{Nf}[b(\bar{g}, 1) - f(x^{k+1}, 0) - (\alpha x^k, 0)] \\ &= (-\alpha x^k, 0).\end{aligned}$$

Observing that if $\delta(a) < k$ (including the case $a = 0$) then a makes no contribution to the value of α, we have proved the following.

Lemma 4.1: Let Nf denote normal form relative to the basis $\{(\bar{g}, 1), (x^{k+1}, 0)\}$ of $\boldsymbol{M}_{k+1}$ (and the order $<_{\delta(\bar{g})}$). Then, for any $(a, b) \in \boldsymbol{M}_k$

$$\mathrm{Nf}(a, b) = (-\alpha x^k, 0)$$

where α is the coefficient of x^k in the expansion of $bg - a$. If $\delta(a) < k$ then $bg - a$ may be replaced by bg.

This leads to an inductive technique as follows. Let

$$\mathcal{B}_k = \{(a_1, b_1), (a_2, b_2)\}$$

be a Gröbner basis for $\boldsymbol{M}_k$ relative to $<_r$ with (a_1, b_1) the minimal element. If $(a_1, b_1) \in \boldsymbol{M}_{k+1}$ then set

$$(a_1', b_1') := (a_1, b_1)$$

and

$$(a_2', b_2') := (xa_2, xb_2).$$

The latter element is clearly in $\boldsymbol{M}_{k+1}$. Otherwise, the coefficient α_1 of x^k in the expansion of $b_1 g - a_1$ is nonzero and, defining α_2 similarly, it is immediate from the previous lemma that

$$\begin{aligned}&\mathrm{Nf}\left[(a_2, b_2) - \frac{\alpha_2}{\alpha_1}(a_1, b_1)\right] \\ &\quad = (-\alpha_2 x^k, 0) - \frac{\alpha_2}{\alpha_1}(-\alpha_1 x^k, 0) \\ &\quad = (0, 0)\end{aligned}$$

so

$$(a_2', b_2') := (a_2, b_2) - (\alpha_2/\alpha_1)(a_1, b_1) \in \boldsymbol{M}_{k+1}.$$

Then define

$$(a_1', b_1') := (xa_1, xb_1).$$

We shall prove that in both cases the updated set

$$\{(a_1', b_1'), (a_2', b_2')\}$$

is a basis of $\boldsymbol{M}_{k+1}$ with leading terms (relative to $<_r$) on opposite sides; by Corollary 2.3 ii) it is a Gröbner basis.

Theorem 4.2: Let

$$\mathcal{B} = \{(a_1, b_1), (a_2, b_2)\}$$

be a Gröbner basis relative to $<_r$ of $\boldsymbol{M}_k$ with (a_1, b_1) minimal and let α_i be the coefficient of x^k in the expansion of $b_i g - a_i$ for $i = 1, 2$. Let

$$\mathcal{B}' = \{(a_1', b_1'), (a_2', b_2')\}$$

be defined as follows:

$$\text{if } \alpha_1 = 0 \text{ then } \begin{cases}(a_1', b_1') = (a_1, b_1) \\ (a_2', b_2') = (xa_2, xb_2)\end{cases}$$

$$\text{if } \alpha_1 \neq 0 \text{ then } \begin{cases}(a_1', b_1') = (xa_1, xb_1) \\ (a_2', b_2') = \left(a_1 - \dfrac{\alpha_2}{\alpha_1} a_2, b_1 - \dfrac{\alpha_2}{\alpha_1} b_2\right).\end{cases}$$

Then $\mathcal{B}'$ is a Gröbner basis of $\boldsymbol{M}_{k+1}$ relative to $<_r$.

Proof: First observe that the leading terms of the elements of $\mathcal{B}'$ are on the same sides as those of the corresponding elements of $\mathcal{B}$, since neither multiplication by x nor subtraction of a scalar multiple of the minimal element causes a change. Thus to apply Corollary 2.3 ii), we need only prove that $\mathcal{B}'$ is a basis of $\boldsymbol{M}_{k+1}$; for this it is sufficient to observe that

$$\{(\bar{g}, 1), (x^{k+1}, 0)\} \subseteq \langle \mathcal{B}' \rangle$$

where the bar denotes reduction modulo x^{k+1}.

Define u_i, v_i by the equations

$$\begin{aligned}(g_{k-1}x^{k-1} + \cdots + g_0, 1) &= u_1(a_1, b_1) + u_2(a_2, b_2) \\ (x^k, 0) &= v_1(a_1, b_1) + v_2(a_2, b_2).\end{aligned}$$

Then, taking the case $\alpha_1 \neq 0$ first, we have

$$\begin{aligned}(x^{k+1}, 0) &= xv_1(a_1, b_1) + xv_2(a_2, b_2) \\ &= \left(v_1 + \frac{\alpha_2}{\alpha_1} v_2\right)(xa_1, xb_1) \\ &\quad + xv_2\left(a_2 - \frac{\alpha_2}{\alpha_1} a_1, b_2 - \frac{\alpha_2}{\alpha_1} b_1\right) \\ &= \left(v_1 + \frac{\alpha_2}{\alpha_1} v_2\right)(a_1', b_1') + xv_2(a_2', b_2')\end{aligned}$$

so $(x^{k+1}, 0) \in \langle \mathcal{B}' \rangle$. On the other hand,

$$\begin{aligned}(\bar{g}, 1) &= g_k(x^k, 0) + (g_{k-1}x^{k-1} + \cdots + g_0, 1) \\ &= g_k[v_1(a_1, b_1) + v_2(a_2, b_2)] \\ &\quad + u_1(a_1, b_1) + u_2(a_2, b_2) \\ &= (g_k v_1 + u_1)(a_1, b_1) + (g_k v_2 + u_2) \\ &\quad \cdot \left(a_2 - \frac{\alpha_2}{\alpha_1} a_1, b_2 - \frac{\alpha_2}{\alpha_1} b_1\right) \\ &\quad + \frac{\alpha_2}{\alpha_1}(g_k v_2 + u_2)(a_1, b_1) \\ &= \left[(g_k v_1 + u_1) + \frac{\alpha_2}{\alpha_1}(g_k v_2 + u_2)\right](a_1, b_1) \\ &\quad + (g_k v_2 + u_2)\left(a_2 - \frac{\alpha_2}{\alpha_1} a_1, b_2 - \frac{\alpha_2}{\alpha_1} b_1\right) \\ &= \frac{1}{x}\left[(g_k v_1 + u_1) + \frac{\alpha_2}{\alpha_1}(g_k v_2 + u_2)\right](a_1', b_1') \\ &\quad + (g_k v_2 + u_2)(a_2', b_2').\end{aligned}$$

It remains to show that

$$\frac{1}{x}\left[(g_k v_1 + u_1) + \frac{\alpha_2}{\alpha_1}(g_k v_2 + u_2)\right]$$

is a polynomial, or in other words, that

$$(g_k v_1 + u_1) + (\alpha_2/\alpha_1)(g_k v_2 + u_2)$$

is divisible by x. This is proved in the lemma that follows.

In the alternative case $\alpha_1 = 0$ and it straightforward to show along similar lines that $(x^{k+1}, 0) \in \mathcal{B}'$, while

$$(\overline{g}, 1) = (g_k v_1 + u_1)(a_1', b_1') + \frac{1}{x}(g_k v_2 + u_2)(a_2', b_2').$$

In this case, we need to prove that $g_k v_2 + u_2$ is divisible by x: again this will follow from the lemma below. This completes the proof of the theorem. □

Lemma 4.3: With the notation above

$$\alpha_1(g_k v_1 + u_1) + \alpha_2(g_k v_2 + u_2)$$

is divisible by x.

Proof: We have the following equations:

$$(g_k v_1 + u_1)a_1 + (g_k v_2 + u_2)a_2 = \overline{g}$$
$$(g_k v_1 + u_1)b_1\overline{g} + (g_k v_2 + u_2)b_2\overline{g} = \overline{g}.$$

It follows that

$$(g_k v_1+u_1)(b_1\overline{g}-a_1)+(g_k v_2+u_2)(b_2\overline{g}-a_2)=0$$
$$(g_k v_1+u_1)(b_1 g-a_1)+(g_k v_2+u_2)(b_2 g-a_2)\equiv 0 \bmod x^{k+1}$$
$$(g_k v_1+u_1)(\alpha_1 x^k)+(g_k v_2+u_2)(\alpha_2 x^k)\equiv 0 \bmod x^{k+1}.$$

Thus

$$[\alpha_1(g_k v_1 + u_1) + \alpha_2(g_k v_2 + u_2)]x^k \equiv 0 \bmod x^{k+1}$$

and the lemma follows. □

Remarks 4.4:

1) We note that the lemma covers the case

$$\alpha_1(g_k v_1 + u_1) + \alpha_2(g_k v_2 + u_2) = 0.$$

This can arise in practice.

2) An alternative proof of the theorem can be given using Corollary 2.3 i).

If $(a, b) \in \boldsymbol{M}_k$ and α is the coefficient of x^k in the expansion of $bg - a$, then α is also the coefficient of x^{k+1} in the expansion of $xbg - xa$. This means that, under both definitions of $\mathcal{B}'$, only one new coefficient needs to be calculated. The next lemma shows that the polynomials a_i are not required except for the initial element $(1, 0)$ and any subsequent multiples $(x^k, 0)$, for which the value of α is -1. Thus in our algorithms we can always obtain the values of α_i as coefficients in the expansions of $b_i g$.

Lemma 4.5: If (a, b) is an element of the basis of $\boldsymbol{M}_k$ generated by the procedure described above then either $(a, b) = (x^k, 0)$ or $\delta a < k$.

Proof: It is clear that all the components of the basis elements have degree at most k. A first component can only have degree k if its degree has risen by 1 in every iteration, that is, if the corresponding basis element started as $(1, 0)$ and was multiplied by x in every iteration. □

We now have the following algorithm, which, for the sake of clarity, is stated in a slightly redundant form. The Boolean variable "$\boldsymbol{L}$" (left) is true if the leading term of the minimal element in the current basis is on the left; otherwise, it is false. The function "swap" interchanges the values of $\{a_1, b_1, \alpha_1\}$ and $\{a_2, b_2, \alpha_2\}$ so that (a_1, b_1) is always the minimal element. We commence with the basis $\{(1, 0), (0, 1)\}$ of $\boldsymbol{M}_0$, so initially the values of α_i are obtained from (the first components of) the normal forms of $(1, 0)$ and $(0, 1)$ relative to the basis $\{(g_0, 1), (x, 0)\}$ of $\boldsymbol{M}_1$. Thus $\alpha_1 = -1$ and $\alpha_2 = g_0$. We denote the coefficient of x^k in the polynomial f by $[f]_k$.

Algorithm 4.6:

Input: g, n, ℓ, m
Output: $(a, b) \in \boldsymbol{M}$ satisfying (3)
Initialize:
$(a_1, b_1) := (1, 0)$; $\alpha_1 := -1$;
$(a_2, b_2) := (0, 1)$; $\alpha_2 := g_0$;
$r := \ell - m$; $k := 0$
IF $r \geq 0$ THEN
  $\boldsymbol{L}$:=true
ELSE
  swap
  $\boldsymbol{L}$:=false
WHILE $k < n$ DO
  $k := k + 1$
  IF $\alpha_1 \neq 0$ THEN
    $(a_2, b_2) := (a_2 - (\alpha_2/\alpha_1)a_1, b_2 - (\alpha_2/\alpha_1)b_1)$
    $(a_1, b_1) := (xa_1, xb_1)$
    $\alpha_2 := [b_2 g]_k$
    IF $\boldsymbol{L}$=true AND $\delta(a_1) > \delta(b_2) + r$ THEN (*)
      swap
      $\boldsymbol{L} :=$ false
    IF $\boldsymbol{L}$ = false AND $\delta(a_2) \leq \delta(b_1) + r$ THEN (*)
      swap
      $\boldsymbol{L} :=$ true
  ELSE
    $(a_2, b_2) := (xa_2, xb_2)$
    $\alpha_1 := [b_1 g]_k$
$(a, b) := \frac{1}{\mathrm{Lc}\,(a_1, b_1)}(a_1, b_1)$

□

Obviously, this algorithm is not in optimal form. First, the swap function would not be implemented in hardware as a physical interchange of the values of the appropriate variables; rather, the registers containing the basis elements would have identical structure and a switch would indicate which stored the minimal element at each stage. This symmetry can be more clearly displayed by relabeling the basis elements (a_0, b_0), (a_1, b_1) and replacing the swap function by a Boolean variable i whose value is the subscript of the minimal element. We denote the complement of i by $\overline{i}$.

Second, the value of $\boldsymbol{L}$ is directly related to the degrees of the components of the basis elements. We can therefore amal-

gamate the tests on the instructions marked $(*)$ by introducing a single new integer control variable d equal to

$$1 - \delta a_i + \delta b_{\bar{i}} + r = 1 - p + q + r$$

when $i = 0$ and

$$\delta a_{\bar{i}} - \delta b_i - r = p - q - r$$

when $i = 1$ and whose value changes by $-1, 0, 1$ at each pass. The parameter d is only tested for equality with zero.

Finally, we introduce a Boolean subscript j to mark which coefficient needs to be calculated at each stage. Initially, $j = 1$ so we do not initialize α_1 (it can be given an arbitrary value if required).

Algorithm 4.7:

Input: g, ℓ, m, n
Output: $(a, b) \in \boldsymbol{M}$ satisfying (3)
Initialize:
$(a_0, b_0) := (1, 0)$; $(a_1, b_1) := (0, 1)$; $\alpha_0 := -1$; $j := 1$
$r := \ell - m$; $k := 0$;
IF $r \geq 0$ THEN
  $i := 0$
  $d := 1 + r$
ELSE
  $i := 1$
  $d := -r$
WHILE $k < n$ DO
  $\alpha_j := [b_j g]_k$
  $k := k + 1$
  IF $\alpha_i \neq 0$ THEN
    $(a_{\bar{i}}, b_{\bar{i}}) := (a_{\bar{i}} - (\alpha_{\bar{i}}/\alpha_i)a_i, b_{\bar{i}} - (\alpha_{\bar{i}}/\alpha_i)b_i)$
    $(a_i, b_i) := (xa_i, xb_i)$
    $j := \bar{i}$
    $d := d - 1$ (†)
    IF $d = 0$ THEN
      $i := \bar{i}$
      $d := 1$ (‡)
  ELSE
    $(a_{\bar{i}}, b_{\bar{i}}) := (xa_{\bar{i}}, xb_{\bar{i}})$
    $j := i$
    $d := d + 1$
$(a, b) := \frac{1}{\mathrm{LC}\,(a_i, b_i)}(a_i, b_i)$.

Proof: First we verify that the calculation of the coefficients α_j is correct. This is obvious for $k = 0$. Now observe that inside the main loop the coefficient that does *not* need to be calculated at the next pass is that associated with a basis element that has been multiplied by x. In each of the two branches j is set to the value corresponding to the other basis element.

Initially $(a_0, b_0) = (1, 0)$ has leading term on the left and $(a_1, b_1) = (0, 1)$ has leading term on the right and $p = q = 0$. At each pass through the main loop each basis element is either multiplied by x or has subtracted from it a scalar multiple of the minimal element. As a consequence the leading term of (a_0, b_0) is always on the left and that of (a_1, b_1) is always on the right.

If $0 \leq 0 + r = r$ then initially $i = 0$ and

$$d = 1 - 0 + 0 + r = 1 + r$$

while if $0 > 0 + r = r$ then $i = 1$ and

$$d = 0 - 0 - r = -r.$$

Note that in either case $d \geq 1$. Now suppose $d \geq 1$ at the start of a pass through the main loop and use primes to represent new values of the variables. Consider first the ELSE path: if $i = 0$ then $p' = p$, $q' = q + 1$ so

$$d' = 1 - p' + q' + r = d + 1$$

while if $i = 1$ then $p' = p + 1$, $q' = q$ so

$$d' = p' - q - r = d + 1.$$

This verifies the correct assignment of the value of d' in this path. A similar argument shows that at (†) in the IF path the value of d' is $d-1$ and that this matches the definition *provided* $d' > 0$. However, if $d' = 0$ at (†) then if $i = 0$ we have

$$d' = 1 - p' + q' + r = 0$$

so that $p' > q' + r$. This implies that (a_1, b_1) is now the minimal element so the value of i is changed to $i' = \bar{i} = 1$ and that of d' to $p' - q' + r = 1$. A similar argument shows that the new values i', d' are correctly assigned in the case that $i = 1$.

This completes the proof of the algorithm. □

We continue by considering an example that illustrates the case $r = -1$ appropriate to the solution of (1).

Example 4.8: Let $1, 0, 0, 0, -1, 1, 0, 0, 1, -2, \cdots$ be the initial terms of a linear recurring sequence over $\boldsymbol{Q}$ whose minimal polynomial has degree at most 5. Then

$$g = h = 1 - x^4 + x^5 + x^8 - 2x^9$$

$n = 2t = 10$, and the computations are shown in Table III (with the values recorded just before the instruction $k := k+1$ and at output).

Thus since the final value of i is 1, the required minimal polynomial is the reciprocal of b_1, that is, $x^5 + x - 1$. □

B. The Case $r \geq 0$: Comparison with BM

Algorithms 4.6 and 4.7 are intimately connected with BM. Indeed, [2, Theorem 7.43] shows that Berlekamp's algorithm also determines bases of a sequence of solution modules. It is easy to see from examples that BM does not produce a reduced solution at each stage. Our algorithms do not produce reduced bases either, since we are only interested in the minimal element; if required the reduced basis can be obtained at the output stage.

Furthermore, the approach taken in the usual implementations of BM is to calculate only the right-hand component b of the minimal element, deriving a as a consequence. This modification may also be carried out in our algorithms: since the a_i, $a_{\bar{i}}$ have no role in the computation of the values of α_i, they can simply be omitted. We illustrate this in the following algorithm which deals with the case $r \geq 0$ (appropriate

TABLE III
EXAMPLE 4.8

k	(a_0, b_0)	(a_1, b_1)	α_0	α_1	i	j	d
0	$(1, 0)$	$(0, 1)$	-1	1	1	1	1
1	$(1, 1)$	$(0, x)$	0	1	0	0	1
2	$(1, 1)$	$(0, x^2)$	0	1	0	0	2
3	$(1, 1)$	$(0, x^3)$	0	1	0	0	3
4	$(1, 1)$	$(0, x^4)$	-1	1	0	0	4
5	(x, x)	$(1, x^4 + 1)$	-1	1	0	1	3
6	(x^2, x^2)	$(x + 1, x^4 + x + 1)$	-1	1	0	1	2
7	(x^3, x^3)	$(x^2 + x + 1, x^4 + x^2 + x + 1)$	-1	1	0	1	1
8	(x^4, x^4)	$(x^3 + x^2 + x + 1,$ $x^4 + x^3 + x^2 + x + 1)$	-1	1	1	1	1
9	$(x^4 + x^3 + x^2 + x + 1,$ $2x^4 + x^3 + x^2 + x + 1)$	$(x^4 + x^3 + x^2 + x,$ $x^5 + x^4 + x^3 + x^2 + x)$	1	1	0	0	1
out	$(x^5 + x^4 + x^3 + x^2 + x,$ $2x^5 + x^4 + x^3 + x^2 + x)$	$(-1, x^5 - x^4 - 1)$	1	1	1	1	1

to errors-and-erasures decoding). This is of special interest because it follows from the argument of 3.1 that when $r \geq 0$ the set

$$\{(a_1, b_1) = (g_0 + g_1 x + \cdots + g_r x^r, 1), (a_0, b_0) = (x^{r+1}, 0)\}$$

is already a Gröbner basis of $\boldsymbol{M}_{r+1}$ with respect to $<_r$. Hence we can begin at that step and need only carry out the $n - r - 1$ iterations to obtain bases of $\boldsymbol{M}_{r+2}, \cdots, \boldsymbol{M}_n$. Note that we have retained the convention that (a_0, b_0) has leading term on the left and (a_1, b_1) has leading term on the right. Thus initially $\alpha_0 = -1$, $i = 1$, $j = 1$, and

$$d = p - q - r = r + 1 - 0 - r = 1.$$

Also, with k initialized to 0, and taking values up to $n - r - 1$, the α_j that needs to be calculated is the coefficient of x^{k+r+1} in $b_j g$.

Algorithm 4.9:

Input: g, ℓ, m, n with $\ell \geq m$
Output: $(a, b) \in \boldsymbol{M}$ satisfying (3)
Initialize:
$b_0 := 0$; $b_1 := 1$; $\alpha_0 := -1$;
$i := 1$; $j := 1$; $d := 1$; $r := \ell - m$; $k := 0$
WHILE $k < n - r - 1$ DO
  $\alpha_j := [b_j g]_{k+r+1}$
  $k := k + 1$
  IF $\alpha_i \neq 0$ THEN
    $b_{\bar{i}} := b_{\bar{i}} - (\alpha_{\bar{i}}/\alpha_i) b_i$
    $b_i = x b_i$
    $j := \bar{i}$
    $d := d - 1$
    IF $d = 0$ THEN
      $i := \bar{i}$
      $d := 1$
  ELSE
    $b_{\bar{i}} := x b_{\bar{i}}$
    $j := i$
    $d := d + 1$
  $a_i := b_i g \bmod x^n$
  $(a, b) := \frac{1}{\mathrm{Lc}(a_i, b_i)} (a_i, b_i)$
□

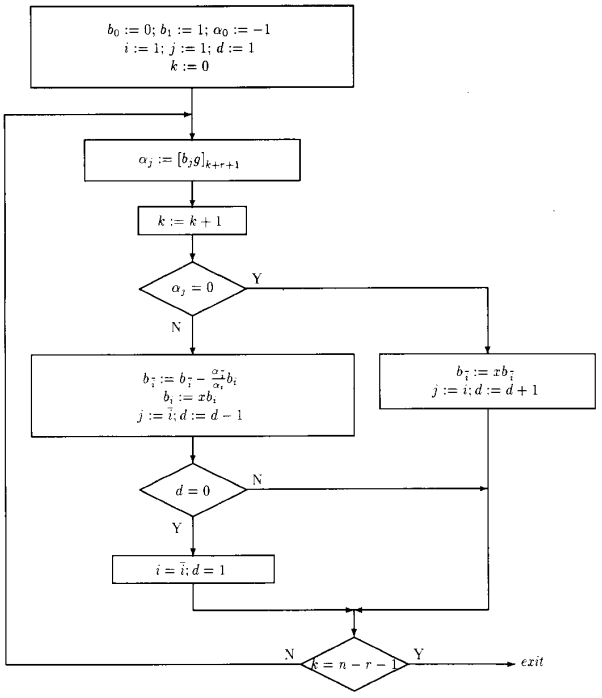

Fig. 1. Algorithm 4.9.

Algorithm 4.9 may be compared directly to BM [4, p. 377, for example] using the flow diagram in Fig. 1. (To obtain the version corresponding to Berlekamp's equation the value of r should be set to 0 and that of n to $2t + 1$.)

We make the following observations concerning this algorithm.

1) It is more symmetrical than BM in its treatment of the two polynomials generated at each stage. This leads to fewer reassignments of polynomials: we have either one reassignment and a shift, or just a shift, whereas BM has either three reassignments, or two reassignments and a shift, or just a shift.

TABLE IV
EXAMPLE 4.10

k	b_0	b_1	α_0	α_1	i	j	d
0	0	1	1	β^3	1	1	1
1	β^{12}	x	1	β^3	0	0	1
2	$\beta^{12}x$	$x+\beta^{10}$	β^5	β^4	1	1	1
3	$\beta^{13}x+\beta^{11}$	$x^2+\beta^{10}x$	β^5	β^4	0	0	1
out	$\beta^{13}x^2+\beta^{11}x$	$x^2+\beta x+\beta^6$	β^9	β^4	1	1	1

2) When $\alpha_i \neq 0$ one polynomial is multiplied by α_i^{-}/α_i. This is half as many multiplications as in BM where one polynomial is multiplied by the "discrepancy" Δ and the other by $1/\Delta$.
3) The control parameter d changes in value by 1 at each pass and is only tested for zero, whereas BM requires an evaluation/comparison test of the form "$2L < m$?"

It follows from this analysis that Algorithms 4.7 and 4.9 are well-suited to practical implementation, and that they improve somewhat on BM in terms of hardware and computational overhead.

Our final example deals with the case $r = 2$ and solves the errors-and-erasures problem [2], [3], [16], [34] for a specific RS code.

Example 4.10: Consider the $(15, 9)$ RS code over GF (16) with generator polynomial

$$x^6 + \beta^{10}x^5 + \beta^{14}x^4 + \beta^4x^3 + \beta^6x^2 + \beta^9x + \beta^6$$

where $\beta^4 + \beta + 1 = 0$, and suppose that the received word is

$$\beta x^{14} + x^{13} + \beta^2x^{11} + \beta^4x^{10} + \beta^2x^9 + \beta^5x^8 + x^7 + \beta x^6 + \beta^7x^5 + \beta^{12}x^4 + \beta^5x^3 + \beta^8x^2 + \beta^4$$

with erasures declared in positions corresponding to β^{11}, β^2 (we choose to keep the received values in the erasure positions rather than setting them to zero.) Then the syndrome polynomial is

$$\beta^{14}x^5 + x^4 + \beta^2x^3 + \beta^5x^2 + \beta^5x + \beta^{12}$$

the erasure locator is

$$\beta^{13}x^2 + \beta^9x + 1$$

and the modified syndrome polynomial is

$$\beta x^6 + \beta^{10}x^5 + \beta^8x^4 + \beta^3x^3 + \beta^9x^2 + \beta^{12}x + 1.$$

With $r = 2$ the computations proceed as in Table IV.

Since the final value of i is 1, the output from the algorithm is

$$\beta^{-6}(\beta x^3 + \beta^{13}x^2 + \beta^9x + \beta^6,\ x^2 + \beta x + \beta^6) = (\beta^{10}x^3 + \beta^7x^2 + \beta^3x + 1,\ \beta^9x^2 + \beta^{10}x + 1).$$

The inverse roots of b give error positions corresponding to β, β^8 and a standard argument provides the erasure and error values 1, β, β^{10}, β^2 in positions β, β^2, β^8, β^{11}, respectively. □

V. CONCLUSIONS

Using the theory of Gröbner bases of polynomial modules we have developed a unified theoretical and practical approach to the solution of the congruence $a \equiv bg \bmod x^n$ for a range of degree conditions on the required solution. This leads to algorithms of the same forms as the Peterson–Gorenstein–Zierler, the extended Euclidean, and the Berlekamp–Massey algorithms. These new algorithms are conceptually simple and all derived from the same fundamental process of converting a Gröbner basis of the solution module relative to one term order into a Gröbner basis relative to another order. The required solution is the unique minimal element relative to the new term order and as such it appears in the new Gröbner basis.

REFERENCES

[1] T. Becker and V. Weispfenning, *Gröbner Bases: A Computational Approach to Commutative Algebra.* New York: Springer-Verlag, 1993.
[2] E. R. Berlekamp, *Algebraic Coding Theory.* New York: McGraw-Hill, 1968.
[3] R. E. Blahut, *Theory and Practice of Error Control Coding.* Reading, MA: Addison-Wesley, 1983.
[4] ——, *Fast Algorithms for Digital Signal Processing.* Reading, MA: Addison-Wesley, 1985.
[5] R. P. Brent, F. G. Gustavson, and D. Y. Y. Yun, "Fast solution of Toeplitz systems of equations and the computation of Padé approximants," *J. Algorithms*, vol. 1, pp. 259–295, 1990.
[6] B. Buchberger, "Gröbner bases: an algorithmic method in polynomial ideal theory," in *Multidimensional Systems Theory*, N. K. Bose, Ed. Dordrecht, The Netherlands: Reidel, 1985, pp. 184–232.
[7] P. Camion, "An iterative Euclidean algorithm," in *Lecture Notes in Computer Science.* vol. 365, L. Huguet and A. Poli, Eds. New York: Springer-Verlag, 1987, pp. 88–128.
[8] X. Chen, I. S. Reed, T. Helleseth, and T. K. Truong, "Use of Gröbner bases to decode binary cyclic codes up to the true minimum distance," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1654–1661, 1994.
[9] ——, "General principles for algebraic decoding of cyclic codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1661–1663, 1994.
[10] A. B. Cooper, "Finding BCH error locator polynomials in one step," *Electron. Lett.*, vol. 27, no. 24, p. 2090, 1991.
[11] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms.* New York: Springer-Verlag, 1992.
[12] J. L. Dornstetter, "On the equivalence between Berlekamp's algorithm and Euclid's algorithm," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 428–431, 1987.
[13] W. L. Eastman, "Inside Euclid's algorithm," in *Coding Theory and Design Theory*, vol. I, D. Ray-Chaudhuri, Ed. New York: Springer-Verlag, 1990.
[14] J. C. Faugère, P. Gianni, D. Lazard, and T. Mora, "Efficient computation of zero-dimensional Gröbner bases by change of ordering," *J. Symb. Comp.*, vol. 16, pp. 329–344, 1993.
[15] P. Fitzpatrick, "Decoding BCH codes by canonical choice in $\boldsymbol{F}[x] \times \boldsymbol{F}[x]$," in *Proc. IMA Conf. on Coding and Cryptography* (Cirencester, UK, Dec. 1993), to appear.
[16] ——, "New time-domain errors-and-erasures decoding algorithm for BCH codes," *Electron. Lett.*, vol. 30, no. 2, pp. 110–111, 1994.
[17] ——, "Change of term order algorithms in certain polynomial modules," Department of Mathematics, University College, Cork, Tech. Rep., July 1993.
[18] P. Fitzpatrick and J. Flynn, "A Gröbner basis technique for Padé approximation," *J. Symb. Comp.*, vol. 13, pp. 133–138, 1992.
[19] P. Fitzpatrick and G. H. Norton, "Linear recurring sequences and an extended subresultant algorithm," in *Lecture Notes in Computer Science*, vol. 388, G. Cohen and J. Wolfmann, Eds. New York, Berlin: Springer-Verlag, 1989, pp. 232–243.
[20] ——, "The Berlekamp–Massey algorithm and linear recurring sequences over a factorial domain," *Appl. Alg. in Eng., Comm., and Comp.*, to appear.
[21] K. O. Geddes, S. R. Czapor, and G. Labahn, *Algorithms for Computer Algebra.* Dordrecht, Boston: Kluwer, 1992.
[22] D. C. Gorenstein and N. Zierler, "A class of error correcting codes in p^m symbols," *J. Soc. Indust. Appl. Math.*, vol. 9, pp. 207–214, 1961.

[23] H. J. Helgert, "Decoding of alternant codes," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 513–514, 1977.
[24] S. Jennings, "A Gröbner basis view of the Welch–Berlekamp algorithm for Reed–Solomon codes," submitted for publication, Sept. 1994.
[25] R. E. Kalman, "On partial realizations, transfer functions, and canonical forms," *Acta Poly. Scand. Math. Comput.*, vol. 31, pp. 9–32, 1979.
[26] J. L. Massey, "Shift register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 122–127, 1969.
[27] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1977.
[28] R. J. McEliece, *The Theory of Information and Coding*. Reading, MA: Addison-Wesley, 1977.
[29] R. J. McEliece and J. B. Shearer, "A property of Euclid's algorithm and an application to Padé approximation," *SIAM J. Appl. Math.*, vol. 34, pp. 611–615, 1978.
[30] W. H. Mills, "Continued fractions and linear recurrences," *Math. Comp.*, vol. 29, pp. 173–180, 1975.
[31] N. Patterson, "The algebraic decoding of Goppa codes," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 203–207, 1975.
[32] W. W. Peterson, "Encoding and error-correction procedures for the Bose–Chaudhuri codes," *IEEE Trans. Inform. Theory*, vol. IT-6, pp. 459–470, 1960.
[33] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed. Boston, MA: MIT Press, 1972.
[34] T. K. Truong, W. L. Eastman, I. S. Reed, and I. S. Hsu, "Simplified procedure for correcting both errors and erasures of Reed–Solomon code using Euclidean algorithm," *Proc. Inst. Elec. Eng.*, pt. E, vol. 135, pp. 318–324, 1988.
[35] Y. Sugiyama, M. Hirasawa, S. Shigeichi, and T. Namekawa, "A method for solving the key equation for decoding Goppa codes," *Inform. Contr.*, vol. 27, pp. 87–99, 1975.
[36] L. R. Welch and R. A. Scholtz, "Continued fractions and Berlekamp's algorithm," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 19–27, 1979.