Gröbner basis solutions of constrained interpolation problems

Henry O'Keeffe, Patrick Fitzpatrick
Department of Mathematics
University College Cork
Cork, Ireland
Email: {hok,fitzpat}@ucc.ie

July 2001

Abstract

This paper extends previous work of the authors on recursive Gröbner basis techniques in coding theory, Padé approximation, partial realization, interpolation, and modeling discrete-time behaviours. We present a general algorithm, applicable to a wide range of constrained interpolation problems in coding theory and systems theory, including list decoding and M–Padé approximation..

KEYWORDS: Gröbner basis, interpolation, coding theory, systems theory, M-Padé, list decoding, Sudan's algorithm, soft-decision decoding

1 Introduction

Let F be a field and A the ring of polynomials $F[x_1, \ldots, x_n]$ in n indeterminates over F. We present a recursive approach to deriving Gröbner bases for submodules of the free module A^q. In [16], we developed a general recursive algorithm for solving systems of congruences of the form

$$\sum_{i=1}^{q} b_i h_{ik} \equiv 0 \bmod I^{(k)}, k = 1, \ldots, p \tag{1}$$

where $h_{ik} \in A$ and the $I^{(k)}$ are ideals in A. The vectors $\mathbf{b} = (b_1, \ldots, b_q)$ that solve such a system form a submodule of A^q. This extended work of Fitzpatrick

[8, 9, 11] where algorithms were developed to deal with special cases, such as Padé approximation, interpolation, partial realization, and decoding alternant (error correcting) codes. These applications were extended in [16] to include the modeling of discrete-time behaviours.

In this paper we further extend our techniques to congruences of the form

$$H^{(k)}(\mathbf{b}) \equiv 0 \bmod M^{(k)}, k = 1, \ldots, p \tag{2}$$

where $\mathbf{b} \in A^q$, $M^{(k)}$ are A–modules, and the $H^{(k)}$ are functions $H^{(k)} : A^q \to M^{(k)}$. The main idea is to extend the results of [16], where the $H^{(k)}$ were essentially linear combinations, to more general homomorphisms and origin shifts. We choose these functions so that, again, the solution vectors form a submodule of A^q. Also, the congruences are now solved modulo modules $M^{(k)}$ rather than ideals. Since the main algorithm produces a Gröbner basis relative to any specified term order, we are free to choose an order that reflects specific problem constraints. We present applications of the algorithm to degree constrained interpolations in systems theory and to list decoding in coding theory. While it is intended primarily as a computational tool, the algorithm can produce closed solutions in certain cases.

In Section 2, we give some background and notation on Gröbner bases, Section 3 is devoted to the general algorithm, Section 4 to a discussion of degree constraints, and Sections 5,6 describe the applications to M–Padé approximation, and list decoding, respectively.

2 Definitions and notation

In this section we introduce some notation and terminology on Gröbner bases. This theory is now well-established and the interested reader is referred to [1, 2, 4], where detailed treatements and comprehensive references to original sources may be found.

The standard basis vector with 1 in position i and 0 elsewhere (and length defined by the context) is denoted $\mathbf{e}_i$. A *term* in A^q is a vector of the type $\mathbf{X} = X\mathbf{e}_i$ where $X = x_1^{t_1} x_2^{t_2} \cdots x_n^{t_n}$ is a *term* in A. Thus a term in A^q is a vector all of whose components are 0 except for one which is a term in A. A *term order* in A^q is a total order $<$ on terms satisfying

(i) $\mathbf{X} < Z\mathbf{X}$ for each term $\mathbf{X}$ in A^q and each term $Z \neq 1$ in A,

(ii) if $\mathbf{X} < \mathbf{Y}$ then $Z\mathbf{X} < Z\mathbf{Y}$ for all terms $\mathbf{X}, \mathbf{Y}$ in A^q and each term Z in A.

An example of such a term order is obtained as follows. Let $<_{\text{lex}}$ denote lexicographic order in A and define $\mathbf{X} = X\mathbf{e_i} < Y\mathbf{e_j}$ if either $i < j$, or $i = j$ and $X <_{\text{lex}} Y$. This is a *position-over-term* (or POT) order.

For non-zero $\mathbf{f} \in A^q$ we may write

$$\mathbf{f} = a_1\mathbf{X}_1 + \cdots + a_r\mathbf{X}_r \tag{3}$$

where the a_i are non-zero constants and the $\mathbf{X}_i$ are terms satisfying $\mathbf{X}_1 > \mathbf{X}_2 > \cdots > \mathbf{X}_r$. The *leading term* $\mathrm{lt}(\mathbf{f})$ of $\mathbf{f}$ is $\mathbf{X}_1$, and the *leading coefficient* $\mathrm{lc}(\mathbf{f})$ is a_1. These definitions are extended to all of A^q by setting $\mathrm{lt}(\mathbf{0}) = \mathbf{0}, \mathrm{lc}(\mathbf{0}) = 0$. For example, with $n = 2, q = 3$, $x_1 <_{\mathrm{lex}} x_2$, and $\mathbf{f} = (x_1^2x_2 + 3x_1^3, x_2 + 1, 2x_1)$, and using the POT order defined above, we find that

$$\mathbf{f} = 2x_1\mathbf{e_3} + x_2\mathbf{e_2} + \mathbf{e_2} + x_1^2x_2\mathbf{e_1} + 3x_1^3\mathbf{e_1}$$

has leading term $x_1\mathbf{e_3}$ and leading coefficient 2. As usual $\mathrm{supp}(\mathbf{f}) = \{\mathbf{X}|\mathbf{X}$ is a term, the coefficient of $\mathbf{X}$ in $\mathbf{f}$ is non-zero$\}$.

If $\mathbf{X} = X\mathbf{e}_i$ and $\mathbf{Y} = Y\mathbf{e}_j$ are terms in A^q we say $\mathbf{X}$ *divides* $\mathbf{Y}$ provided $i = j$ and X divides Y in A, that is, if there is a term Z (the quotient) in A satisfying $Z\mathbf{X} = \mathbf{Y}$. A set of non-zero vectors $\mathcal{G} = \{\mathbf{g}_1, \ldots, \mathbf{g}_r\}$ contained in the submodule M is called a *Gröbner basis* of M if for all $\mathbf{f} \in M$ there exists $i \in \{1, \ldots, r\}$ such that $\mathrm{lt}(\mathbf{g}_i)$ divides $\mathrm{lt}(\mathbf{f})$. In particular, $\mathcal{G}$ is a basis of M.

We say that a Gröbner basis is *strictly ordered* if there are no duplicates among its leading terms (although some may be multiples of others), and its elements are in increasing order of leading term. Any Gröbner basis can be converted to a strictly ordered Gröbner basis by simply deleting any element whose leading term is the same as that of another element (choosing arbitrarily in case of ties). In the algorithms presented here we will use a generic function *ord* which returns a strictly ordered Gröbner basis $\mathrm{ord}(\mathcal{G})$ for any Gröbner basis $\mathcal{G}$ (no difficulty arises as a consequence of any lack of uniqueness). A Gröbner basis is *minimal* if none of its elements has leading term a multiple of the leading term of another of its elements. Again, any Gröbner basis can be converted into a minimal Gröbner basis simply by deleting appropriate elements. It is often more efficient, although not strictly necessary, to define the function *ord* so that it constructs ordered minimal Gröbner bases (see Example 6.2), but in some situations such bases are not appropriate (see Algorithm 6.5, Example 6.6) and so we do not adopt them universally.

Each $\mathbf{f} \in M$ has a *standard representation*, with respect to a Gröbner basis $\mathcal{G} = \{\mathbf{g}_1, \ldots, \mathbf{g}_m\}$, of the form $\mathbf{f} = \sum_{i=1}^m a_i\mathbf{g}_i$ where $a_i \in A$ and $\mathrm{lt}(f_i\mathbf{g}_i) \leq \mathrm{lt}(\mathbf{f})$, $1 \leq i \leq m$. A *minimal* element in a submodule $M \subseteq A^q$ is one whose leading term is least among the elements of M, under the given term order $<$. It is unique up to a constant multiple and must appear in any Gröbner basis relative to $<$.

We require several different types of degree. The *multidegree* $\Delta(X^{\underline{\beta}})$ of a term $X^{\underline{\beta}} = x_1^{\beta_1}x_2^{\beta_2}\ldots x_n^{\beta_n} \in A$, $\beta_j \in \mathbf{N}_0, 1 \leq j \leq n$, is defined by $\Delta(X^{\underline{\beta}}) = \underline{\beta}$. If $<$ is a term order on A, the multidegree of $f \in A$ is given by $\Delta(f) = \Delta(\mathrm{lt}(f))$. More generally, if $<$ is a term order on A^q and $\mathrm{lt}(\mathbf{f}) = X^{\underline{\beta}}\mathbf{e}_i$ then $\Delta(\mathbf{f}) = \underline{\beta}$. Next, if $\underline{\alpha} = (\underline{\alpha}^{(1)}, \ldots, \underline{\alpha}^{(q)}) \in (\mathbf{N}_0^n)^q$, we define $\underline{\lambda} = \underline{\lambda}_{\underline{\alpha}}$ by $X^{\underline{\lambda}} = \mathrm{lcm}\{X^{\underline{\alpha}^{(1)}}, \ldots, X^{\underline{\alpha}^{(q)}}\}$. Then the $\underline{\alpha}$–*modified multidegree* of $\mathbf{f}$, where the leading term of $\mathbf{f}$ is $X^{\underline{\beta}}\mathbf{e}_i$, is defined as $\Delta_{\underline{\alpha}}(\mathbf{f}) = \underline{\beta} + \underline{\lambda} - \underline{\alpha}^{(i)}$. It will often be convenient to identify a term order on A with the corresponding total order on $\mathbf{N}_0^n$, that is, $X^{\underline{\beta}} < X^{\underline{\gamma}}$ if and only if $\underline{\beta} < \underline{\gamma}$, and we adopt this convention without further mention.

Example 2.1 In Section 4, we require a term order, based on the modified multidegree, defined as follows. First select an arbitrary but fixed term order $<_1$ on A and let *tb* denote an arbitrary, fixed tie breaking function on $\{1, \ldots, q\}$. Let $\underline{\alpha} = (\underline{\alpha}^{(1)}, \ldots, \underline{\alpha}^{(q)})$ be fixed, and let $U = X^{\underline{\beta}_U}, V = X^{\underline{\beta}_V}$. Define $<$ by

$$\begin{array}{rcl} U\mathbf{e}_i & < & V\mathbf{e}_j \\ & \text{when} & \Delta_{\underline{\alpha}}(U\mathbf{e}_i) <_1 \Delta_{\underline{\alpha}}(V\mathbf{e}_j) \\ & \text{or} & \Delta_{\underline{\alpha}}(U\mathbf{e}_i) = \Delta_{\underline{\alpha}}(V\mathbf{e}_j) \text{ and } \mathrm{tb}(i,j) = i. \end{array} \tag{4}$$

The proof that this is a term order is given in Lemma 4.3. □

Finally, for use in Section 6, if $f = \sum_{i,j} f_{ij}x^iy^j \in F[x,y]$ and a, b are positive integers we define the (a,b)–*degree* of f by $\Delta_{(a,b)}(f) = \max\{ai + bj | x^iy^j \in \mathrm{supp}(f)\}$, and use the standard term *total* degree for the $(1,1)$–degree (no confusion will arise between these uses of the symbol Δ). Also, in that section we will write ∂g for the degree of a 1–variable polynomial g.

Example 2.2 The (a,b)–degree can be used to construct term orders in $F[x,y]$ in which the terms are first ordered by (a,b)–degree, and then by some suitable tie breaker such as the degree in y. If $(a,b) = (2,3)$ this leads to the following order

$$1 < x < y < x^2 < xy < x^3 < y^2 < x^2y < x^4 < xy^2 < \cdots.$$

□

3 The general algorithm

We first describe the incremental step of our algorithm in the following theorem. This will be applied to submodules $M_\ell, M_{\ell+1}$ of an A–module M, with $M_\ell \supseteq M_{\ell+1}$, such that, for each $s, 1 \le s \le n$, there exists $\beta_s \in F$ satisfying

$$(x_s - \beta_s)M_\ell \subseteq M_{\ell+1}. \tag{5}$$

We also require, for each ℓ, an F–homomorphism

$$\theta_\ell : M_\ell \rightarrow F \tag{6}$$

with $ker(\theta_\ell) = M_{\ell+1}$.

While the theorem and algorithm strongly reflect the corresponding elements in [16], the new setting significantly extends the applicability of the results (as will be seen in Sections 5 and 6). The proof is a fairly straightforward generalization of that in [16], but is included to make this paper relatively self-contained. If $\mathcal{W}$ is an ordered set then $\mathcal{W}[j]$ denotes its j^{th} element.

Theorem 3.1 *Let M be an A–module and let $M_\ell \supseteq M_{\ell+1}$ be submodules of M satisfying* (5) *and* (6) *for suitable β_s, θ_ℓ. Let $H : A^q \rightarrow M$ be an F–linear function such that for each $s, 1 \leq s \leq n$ there exists $\gamma_s \in F$ satisfying*

$$H(x_s\mathbf{b}) = (x_s + \gamma_s)H(\mathbf{b})$$

for all $\mathbf{b} = (b_1, \ldots, b_q) \in A^q$. Let $S \subseteq A^q$ be a submodule satisfying

$$H(\mathbf{b}) \equiv 0 \bmod M_\ell \textit{ for all } \mathbf{b} \in S \tag{7}$$

and let $S' \subseteq S$ be the set of elements satisfying

$$H(\mathbf{b}) \equiv 0 \bmod M_{\ell+1}. \tag{8}$$

Then S' is a submodule of A^q.

If $\mathcal{W}$ is a strictly ordered Gröbner basis of S relative to a term order $<$ then a Gröbner basis $\mathcal{W}'$ of S' relative to $<$ can be constructed as follows

Define $\alpha_j := \theta_\ell(H(\mathcal{W}[j]))$ *for* $1 \leq j \leq \lvert\mathcal{W}\rvert$.
If $\alpha_j = 0$ *for all* j *then*
$\quad\mathcal{W}' = \mathcal{W}$
otherwise
$\quad j^* :=$ *least* j *for which* $\alpha_j \neq 0$
$\quad\mathcal{W}_1 := \{\mathcal{W}_j : j < j^*\}$
$\quad\mathcal{W}_2 := \{(x_s - (\beta_s + \gamma_s))\mathcal{W}[j^*] : 1 \leq k \leq n\}$
$\quad\mathcal{W}_3 := \{\mathcal{W}[j] - (\alpha_j/\alpha_{j^*})\mathcal{W}[j^*] : j > j^*\}$
$\quad\mathcal{W}' := \mathcal{W}_1 \bigcup \mathcal{W}_2 \bigcup \mathcal{W}_3$

Proof. If $\mathbf{b} \in S'$ then $H(x_s\mathbf{b}) = (x_s + \gamma_s)H(\mathbf{b}) \equiv 0 \bmod M_\ell$. Thus, $H(X\mathbf{b}) \in S'$ for any term X. By the F–linearity of H it follows straightforwardly that S' is a submodule.

By definition, $H(\mathcal{W}[j]) \in M_\ell$, so if $\alpha_j = 0$, for all j, then $\mathcal{W} \subseteq S'$ so $S' = S$. Thus suppose some $\alpha_j \neq 0$ and let j^* be as defined. If $j < j^*$ then clearly $\mathcal{W}[j] \in S'$. Next, $H(\mathcal{W}[j^*]) \in M_l$, so $H((x_s - \gamma_s - \beta_s)\mathcal{W}[j^*]) = (x_s - \beta_s)H(\mathcal{W}[j^*]) \in M_{\ell+1}$, by (5), and hence $(x_s - (\beta_s + \gamma_s))\mathcal{W}[j^*] \in S'$. Finally, for $j > j^*$,

$$\begin{aligned} \theta_\ell(H(\mathcal{W}[j] - (\alpha_j/\alpha_{j^*})\mathcal{W}[j^*])) &= \theta_\ell(H(\mathcal{W}[j]) - (\alpha_j/\alpha_{j^*})H(\mathcal{W}[j^*])) \\ &= \theta_\ell(H(\mathcal{W}[j])) - (\alpha_j/\alpha_{j^*})\theta_\ell(H(\mathcal{W}[j^*])) \\ &= \alpha_j - (\alpha_j/\alpha_{j^*})\alpha_{j^*} = 0 \end{aligned}$$

so $\mathcal{W}[j] - (\alpha_j/\alpha_{j^*})\mathcal{W}[j^*] \in S'$ by (6). We have now proved that $\mathcal{W}' \subseteq S'$.

We show that $\mathcal{W}'$ is a Gröbner basis as follows. By assumption, $\mathrm{lt}(\mathcal{W}[i]) \neq \mathrm{lt}(\mathcal{W}[j])$, when $i \neq j$. Now, $\mathrm{lt}((x_s - (\beta_s + \gamma_s))\mathcal{W}[j^*]) = x_s\mathrm{lt}(\mathcal{W}[j^*])$ and $\mathrm{lt}(\mathcal{W}[j] - (\alpha_j/\alpha_{j^*})\mathcal{W}[j^*])) = \mathrm{lt}(\mathcal{W}[j]), j > j^*$. Let $\mathbf{f} \in S' \subseteq S$. Then $\mathrm{lt}(\mathbf{f})$ is divisible by some $\mathrm{lt}(\mathcal{W}[j])$. If $j \neq j^*$ then $\mathrm{lt}(\mathbf{f})$ is divisible by the leading term of an element of $\mathcal{W}'$.

Thus, we may suppose that $\mathrm{lt}(\mathcal{W}[j^*])$ is the *only* leading term of the basis elements $\mathcal{W}[j]$ that divides $\mathrm{lt}(\mathbf{f})$. We show that $x_s\mathrm{lt}(\mathcal{W}[j^*])$ also divides $\mathrm{lt}(\mathbf{f})$ for some s. Consider the standard representation $\mathbf{f} = \sum_{j\in J} f_j\mathcal{W}[j]$, with $f_j \neq 0$, and $J \subseteq \{1, \ldots, |\mathcal{W}|\}$. By definition of this representation, and by the assumption on $\mathrm{lt}(\mathcal{W}[j^*])$, it follows that $j^* \in J$, $\mathrm{lt}(\mathbf{f}) = \mathrm{lt}(f_{j^*}\mathcal{W}[j^*])$, and $\mathrm{lt}(f_j\mathcal{W}[j]) < \mathrm{lt}(\mathbf{f})$ for $j \neq j^*$. Let X_j, $j \in J$, be terms in A such that $\mathrm{lt}(f_j\mathcal{W}[j]) = X_j\mathrm{lt}(\mathcal{W}[j])$. Thus $X_j\mathrm{lt}(\mathcal{W}[j]) < X_{j^*}\mathrm{lt}(\mathcal{W}[j^*])$. Suppose that there is some $j \in J$ with $j > j^*$. If $X_{j^*} = 1$ then

$$X_j\mathrm{lt}(\mathcal{W}[j]) < \mathrm{lt}(\mathcal{W}[j^*]) \leq X_j\mathrm{lt}(\mathcal{W}[j^*])$$

which contradicts the increasing order of $\mathcal{W}$. Hence $X_{j^*} \neq 1$, so $x_s\mathrm{lt}(\mathcal{W}[j^*])$ divides $\mathrm{lt}(\mathbf{f})$ for some s. Otherwise, $J \subseteq \{1, \ldots, j^*\}$ and

$$\mathbf{f} - \sum_{j=1}^{j^*-1} f_j\mathcal{W}[j] = f_{j^*}\mathcal{W}[j^*]$$

lies in S'. Therefore $f_{j^*} \neq 1$ since $\mathcal{W}[j^*] \notin S'$. As a consequence, $X_{j^*} \neq 1$ and again $x_s\mathrm{lt}(\mathcal{W}[j^*])$ divides $\mathrm{lt}(\mathbf{f})$ for some s. This completes the proof. $\square$

We can apply this result, in particular, when H is an A-homomophism (that is, when $\gamma_s = 0$ for all s), and when $M = A$ and $M_\ell, M_{\ell+1}$ are ideals. This is the situation studied in [16], where the functions $H^{(k)}$ were defined with reference to vectors $\mathbf{h}^{(k)} \in A^q$ so that $H^{(k)}(\mathbf{b}) = \mathbf{h}^{(k)}.\mathbf{b}$ (dot product).

The solution set T of (2) is a submodule of A^q provided the functions $H^{(k)}$ satisfy the required conditions (5), (6). In this case we may approach the solution module by a sequence of approximations, as follows. Suppose that for each k we have a descending chain of modules

$$M_0^{(k)}, \ldots, M_\ell^{(k)}, \ldots, M_{N_k}^{(k)} = M^{(k)}$$

and suitable constants β, γ and homomorphisms θ so that each pair $M_\ell^{(k)} \supseteq M_{\ell+1}^{(k)}$, together with the function $H^{(k)}$ satisfies the conditions of Theorem 3.1. Let $(j_1, \ldots, j_p) \in \{1, \ldots, N_1\} \times \cdots \times \{1, \ldots, N_p\}$. We define $T_{(j_1,\ldots,j_p)}$ to be the submodule satisfying

$$H^{(k)}(\mathbf{b}) \equiv 0 \bmod M_{j_k}^{(k)} \text{ for all } k \in \{1, \ldots, p\}. \tag{9}$$

Let $L = \sum_{k=1}^{p} N_k$, and set $T^{(0)} = T_{(0,\ldots,0)}$, $T^{(L)} = T_{(N_1,\ldots,N_p)}$. We will define a descending chain of modules $T^{(0)} \supseteq \cdots \supseteq T^{(L)}$. Beginning with an initial strictly ordered Gröbner basis for $T^{(0)}$ we will use the incremental step to determine a Gröbner basis for the next module in the sequence. If $j_k \leq j'_k$ for all $k \in \{1, \ldots, p\}$ then $T_{(j_1,\ldots,j_p)} \supseteq T_{(j'_1,\ldots,j'_p)}$. A situation, arising frequently in applications, occurs when $M_0^{(k)}$ is the image of $H^{(k)}$ for at least one $k \in \{1, \ldots, p\}$. Then the standard basis of A^q, ordered with respect to $<$, is a suitable initial basis.

Now suppose that we have a strictly ordered Gröbner basis for $T^{(i)} = T_{(j_1,\ldots,j_p)}$. Then, providing $j'_k = j_k + 1$ for exactly one $k \in \{1, \ldots, p\}$, and $j'_k = j_k$ otherwise, the incremental step provides a Gröbner basis for $T^{(i+1)} = T_{(j'_1,\ldots,j'_p)}$, where we select $M_\ell = M_j^{(k)}, M_{\ell+1} = M_{j+1}^{(k)}$ along with the corresponding constants β, γ and the homomorphism θ. The resulting Gröbner basis is then converted into a strictly ordered Gröbner basis by the function *ord*. Thus the solution module $T^{(L)}$ can be approached along different sequences of modules. In Algorithm 3.2 the choice of sequence is implemented by the function *nextmod*.

Algorithm 3.2
Input
functions $H^{(k)}$
constants $\gamma_s^{(k)}, 1 \leq k \leq p, 1 \leq s \leq n$
modules $M_\ell^{(k)}$ and homomorphisms $\theta_\ell^{(k)}, 1 \leq k \leq p, 0 \leq \ell \leq N_k$
constants $\beta_s^{(k)}, 1 \leq k \leq p, 1 \leq s \leq n$
$<$ a term order on A^q
$T^{(0)}$ the initial module
$\mathcal{W}_0$ a strictly ordered Gröbner basis of $T^{(0)}$
Output
$\mathcal{W}$ a strictly ordered Gröbner basis of the submodule $T^{(L)}$
Main Routine
$\mathcal{W} := \mathcal{W}_0$
For *module* from $T^{(0)}$ to $T^{(L)}$
$(k, \theta_\ell) = nextmod(module)$
$\alpha_j := \theta_\ell(H^{(k)}(\mathcal{W}[j]))$ for $j \in [|\mathcal{W}|]$
If $\alpha_j = 0$ for all j then $\mathcal{W}' := \mathcal{W}$
else
$j^* :=$ least j for which $\alpha_j \neq 0$
$\mathcal{W}_1 := \{\mathcal{W}_j : j < j^*\}$
$\mathcal{W}_2 := \{(x_s - (\beta_s^{(k)} + \gamma_s^{(k)}))\mathcal{W}[j^*] : 1 \leq s \leq n\}$
$\mathcal{W}_3 := \{\mathcal{W}[j] - (\alpha_j/\alpha_{j^*})\mathcal{W}[j^*] : j > j^*\}$
$\mathcal{W}' := \mathcal{W}_1 \bigcup \mathcal{W}_2 \bigcup \mathcal{W}_3$
$\mathcal{W} := \mathrm{ord}(\mathcal{W}')$

4 Degree constraints

Many problems arising in practice further constrain the required solutions by imposing limits on their degrees. We present a term order which, when used in conjunction with the incremental step, allows these solutions to be identified.

Let $<_1$ be a term order on A. A term order $<$ on A^q is said to be *compatible* with $<_1$ if $a\mathbf{e}_i < b\mathbf{e}_i$ whenever $a <_1 b$. The following lemma will be used freely in the sequel. (In fact, this can be viewed as an alternative definition of a compatible term order—see [5, p. 341].)

Lemma 4.1 *If* $\mathrm{lt}_1(h)$ *is the leading term of* $h \in A$ *with respect to* $<_1$ *and* $\mathrm{lt}(\mathbf{g})$ *is the leading term of* $\mathbf{g} \in A^q$ *with respect to a term order* $<$*, which is compatible with* $<_1$*, then*

$$\mathrm{lt}(h\mathbf{g}) = \mathrm{lt}_1(h)\mathrm{lt}(\mathbf{g})$$

PROOF. Let $\mathrm{lt}_1(h) = u$, $\mathrm{lt}(\mathbf{g}) = w\mathbf{e}_k$. By the definition of term order, every term $v\mathbf{e}_j \in \mathrm{supp}(\mathbf{g})$ satisfies

$$uv\mathbf{e}_j < uw\mathbf{e}_k.$$

Every term $t \in \mathrm{supp}(h)$ satisfies $t <_1 u$. So, by compatibility,

$$t\mathbf{e}_j < u\mathbf{e}_j$$

and, hence, by the definition of term order,

$$tv\mathbf{e}_j < uv\mathbf{e}_j.$$

Now every term in $h\mathbf{g}$ has the form $t(v\mathbf{e}_j)$ for $t \in \mathrm{supp}(h)$, $v\mathbf{e}_j \in \mathrm{supp}(\mathbf{g})$ and

$$t(v\mathbf{e}_j) = (tv)\mathbf{e}_j < (uv)\mathbf{e}_j <_q (uw)\mathbf{e}_k.$$

So $(uw)\mathbf{e}_k = \mathrm{lt}(h\mathbf{g})$ as required. $\square$

The next example shows that not all term orders on A^q are compatible with some term order on A.

EXAMPLE 4.2 Let $A = F[x, y]$ and $q = 2$. If $<_1$ is any term order with $x < y$ (such as a lexicographic order) and $<_{1'}$ is any term order with $y < x$ (such as a different lexicographic order),we can define a term order $<_2$ on $F[x, y]^2$ by $X_1\mathbf{e}_i <_2 X_2\mathbf{e}_j$ if $(i < j)$ or $(i = j = 1$ and $X_1 <_1 X_2)$ or $(i = j = 2$ and $X_1 <_{1'} X_2)$. Let $h = x + y$ and let $<$ be any term order on A. Since $\mathrm{lt}_2((x+y)\mathbf{e}_1) = y\mathbf{e}_1$ and $\mathrm{lt}_2((x+y)\mathbf{e}_2) = x\mathbf{e}_2$ we see that $<_2$ is not compatible with $<$. $\square$

We now consider the term order defined in Example 2.1.

Lemma 4.3 *The relation $<$ defined in* (4) *is a term order on A^q which is compatible with $<_1$.*

PROOF. Let $\underline{\lambda} = \underline{\lambda}_{\underline{\alpha}}$. The relation $<$ defines a total order on the terms of A^q, because $<_1$ is total order and, in the event of equality at the same component $\mathbf{e}_k$, we have $\underline{\beta}_U + \underline{\lambda} - \underline{\alpha}_k = \underline{\beta}_V + \underline{\lambda} - \underline{\alpha}_k$, so the terms are equal. Next, let $\mathbf{U} = X^{\underline{\beta}_U}\mathbf{e}_k, Z = X^{\underline{\beta}_Z} \neq 1$. Then $\underline{\beta}_U + \underline{\lambda} - \underline{\alpha}^{(k)} <_1 \underline{\beta}_Z + \underline{\beta}_U + \underline{\lambda} - \underline{\alpha}^{(k)}$ so $\mathbf{U} < Z\mathbf{U}$. Finally, if $\mathbf{U} = X^{\underline{\beta}_U}\mathbf{e}_i < \mathbf{V} = X^{\underline{\beta}_V}\mathbf{e}_j$ and $\underline{\beta}_U + \underline{\lambda} - \underline{\alpha}^{(i)} <_1 \underline{\beta}_V + \underline{\lambda} - \underline{\alpha}^{(j)}$ then adding $\underline{\beta}_Z$ to both sides does not change the order, while if $\underline{\beta}_U + \underline{\lambda} - \underline{\alpha}^{(i)} = \underline{\beta}_V + \underline{\lambda} - \underline{\alpha}^{(j)}$ and $\mathrm{tb}(i, j) = i$, then adding $\underline{\beta}_Z$ to both sides preserves the equality and $\mathrm{tb}(i, j) = i$ is unchanged. In either case $Z\mathbf{U} < Z\mathbf{V}$, as required. $\square$

In many applications solutions $\mathbf{f} = (f^{(1)}, \ldots, f^{(q)})$ of (1) are sought, which satisfy the condition

$$\Delta(f^{(i)}) <_1 \underline{\alpha}^{(i)} \text{ for } 1 \leq i \leq q \tag{10}$$

for some fixed term order $<_1$ and vector $\underline{\alpha}$ of multidegrees. This generic problem of finding solution vectors satisfying degree constraints is the natural generalization of the 1–variable case considered in [8, 9]. Such "required solutions" can be determined from a Gröbner basis of the solution module.

Theorem 4.4 *Let $\mathcal{G} = \{\mathbf{g}_1, \ldots, \mathbf{g}_m\}$ be a Gröbner basis, with respect to the term order $<$ defined in* (4)*, of the solution module of* (2)*. A vector $\mathbf{f} \in A^q$ satisfies* (10) *if and only it can be expressed in the form*

$$\mathbf{f} = \sum_{k \in K} a_k \mathbf{g}_k \tag{11}$$

where $K \subseteq \{1, \ldots, m\}$, and $a_k, \mathbf{g}_k$ satisfy $\Delta(a_k) + \Delta_{\underline{\alpha}}(\mathbf{g}_k) \leq_1 \underline{\lambda} = \underline{\lambda}_{\underline{\alpha}}$, for $k \in K$.

Proof. Suppose $\mathbf{f}$ can be expressed as in (11). Then $\mathrm{lt}(\mathbf{f}) \leq \mathrm{lt}(a_k \mathbf{g}_k) = \mathrm{lt}_1(a_k)\mathrm{lt}(\mathbf{g}_k)$ for some $k \in K$. If $\mathrm{lt}(\mathbf{f}) = X^{\underline{\beta}_f}\mathbf{e}_i$ and $\mathrm{lt}(\mathbf{g}_k) = X^{\underline{\beta}_g}\mathbf{e}_j$ then $\underline{\beta}_f + \underline{\lambda} - \underline{\alpha}^{(i)} \leq_1 \Delta(a_k) + \underline{\beta}_g + \underline{\lambda} - \underline{\alpha}^{(j)} = \Delta(a_k) + \Delta_{\underline{\alpha}}(\mathbf{g}_k) \leq_1 \underline{\lambda}$. Since $\mathrm{lt}(f^{(r)}\mathbf{e}_r) \leq \mathrm{lt}(\mathbf{f})$ for all $1 \leq r \leq q$, $\Delta(f^{(r)}) + \underline{\lambda} - \underline{\alpha}^{(r)} \leq_1 \underline{\beta}_f + \underline{\lambda} - \underline{\alpha}^{(i)} \leq_1 \underline{\lambda}$ and hence $\Delta(f^{(r)}) \leq_1 \underline{\alpha}^{(r)}$. Therefore $\mathbf{f}$ is a solution of the required type.

Conversely, suppose $\mathbf{f}$ is a solution of the required type. The standard representation is $\mathbf{f} = \sum_{k \in K} a_k \mathbf{g}_k$ for some $K \subseteq \{1, \ldots, m\}$, where $\mathrm{lt}(a_k \mathbf{g}_k) \leq \mathrm{lt}(\mathbf{f})$ for all $k \in K$. Now $\mathrm{lt}(a_k \mathbf{g}_k) = \mathrm{lt}_1(a_k)\mathrm{lt}(\mathbf{g}_k)$, so $\mathrm{lt}_1(a_k)\mathrm{lt}(\mathbf{g}_k) \leq \mathrm{lt}(\mathbf{f})$. With the notation of the previous paragraph, this gives $\Delta(a_k) + \Delta_{\underline{\alpha}}(\mathbf{g}_k) \leq_1 \underline{\beta}_f + \underline{\lambda} - \underline{\alpha}^{(i)}$. Since, by assumption, $\underline{\beta}_f \leq_1 \underline{\alpha}^{(i)}$, the result follows. □

Corollary 4.5 *Suppose that $\mathcal{G}$ is a strictly ordered Gröbner basis of the solution module of* (2)*. There exists a solution satisfying* (10) *if and only if $\Delta_{\underline{\alpha}}(\mathbf{g}_1) \leq_1 \underline{\lambda}$, and in that case $\mathbf{g}_1$ is such a solution. It is the unique such solution (up to constant multiples) if and only if $\Delta_{\underline{\alpha}}(\mathbf{g}_1) = \underline{\lambda}$.*

5 The extended M–Padé problem

Beckermann and Labahn [3] introduced the extended M–Padé approximation problem, as a generalization of the Padé and Hermite–Padé problems, to address a wide variety of interpolation and approximation problems in a single variable. They used module theoretic methods to parameterize the set of all solutions of the general problem. Further, they developed a method of recursively generating solution bases that enabled them to break a given problem down into two smaller problems of the same type along arbitrary paths. In this section we indicate how our algorithm can be applied to this problem. It is clear that our approach extends beyond the single variable case.

Let $F_0 = \{\beta_1, \ldots, \beta_t\} \subseteq F$ be a set of "knots" or interpolation points. Let $F^{p\times q}[[x]]_{F_0}$ be the set of formal Newton series with coefficients in $F^{p\times q}$, that is, $G \in F^{p\times q}[[x]]_{F_0}$ if for all $\beta_j \in F_0$ and all $r \in \mathbf{N}_0$, the r^{th} derivative of G at β_j is known and is an element of $F^{p\times q}$. The set of polynomials with coefficients in $F^{p\times q}$ is a subset of $F^{p\times q}[[x]]_{F_0}$. A vector $\mathbf{u} = (u_1, \ldots, u_q)$ of monic polynomials all of whose zeros are elements of F_0, is called an *order vector* (with respect to F_0). Let $G \in F^{p\times q}[[x]]_{F_0}$. We seek a basis of the set of polynomial vectors $P \in F^{q\times 1}[x]$ satisfying

$$GP = \operatorname{diag}(u_1, \ldots, u_q)W$$

where the *residual* W lies in $F^{q\times 1}[[x]]_{F_0}$.

Setting $G = (g_{ij})$, $P = (p_j)$ and $W = (w_i)$, this becomes

$$\sum_{i=1}^{q} g_{ki}p_i = u_k w_k, k = 1, \ldots, p.$$

Since u_k is a product of factors $(x - \beta_j)^{n_j}$ this is equivalent to

$$\sum_{i=1}^{q} g_{ki}p_i \equiv 0 \bmod \langle \prod_{j\in J} (x - \beta_j)^{n_j} \rangle \tag{12}$$

for some subset $J \subseteq \{1, \ldots, t\}$. Here g_{k_i}, w_k are Newton series rather than polynomials series. So, while (12) is suggestive of (1), the congruence holds in a ring of Newton series instead of a polynomial ring. It does, however, conform to the requirements of (2). The ideal $M^{(k)} = \langle u_k \rangle$ can be regarded as a module over the polynomial ring. The sum of products of Newton series and polynomials on the left hand side can be viewed as a function from A^q into this module.

Example 5.1 In Example 2.9 from [3] we have $F_0 = \{0\}, p = q = 3$ and

$$\left(\frac{1}{2} + x^2 - x^4\right) p_1 + \left((1 + \sin^4(x^2)\right) p_2 + \left(\frac{1}{\sqrt{1 + x^2}}\right) p_3 \equiv 0 \bmod \langle x^8 \rangle.$$

Since the other two conditions are modulo $\langle 1 \rangle$, they are automatically satisfied. □

To apply Algorithm 3.2, we observe that in each of the congruences, a typical pair of successive modules in the descending sequence has the form $M_\ell = \langle f \rangle \supseteq \langle (x -$

$\beta_s)f\rangle = M_{\ell+1}$. Thus (5) is satisfied. If $g \in M_\ell$ then $g = fw$ for some $w \in F[[x]]_{F_0}$, and expanding w around β_s, we can define θ_ℓ by taking $\theta_\ell(g)$ to be the constant term in the expansion. Thus the problem may be solved using our techniques.

Moreover, we can not only subdivide the problem into subproblems of the same type [3], but also solve any or all them using our algorithm. The ultimate solution can be arrived at exclusively by that algorithm or in combination with other methods. As in [16], the incremental step is much simpler for single variable problems than in the general case, because the position of the leading term of each basis element is not changed, so that the function *ord* has only to move one basis element at each iteration.

However, the types of degree constraint allowed by the methods of [3] are more general than those that we can apply. Essentially, they permit different constraints in different congruences, whereas our algorithm depends on the constraint being the same throughout. For this special type of constraint we can parameterize all solutions (Theorem 4.4). Of course, there are fundamental issues of numerical stability involved in the problems addressed in [3]. We have not yet been able to carry out a full analysis of the numerical properties of our algorithm, but some indications of its performance in a finite precision environment are given in [10].

We conclude this section by indicating some of the theoretical results from [3] that are straightforward consequences of our approach. We follow that paper in assuming, without loss of generality, that $p = q$. First, since there are q elements in the initial (standard) basis, there are q elements in each basis ([3, Theorem 2.6]). Second, if the elements of a basis are arranged in a matrix $\mathbf{P}$ then the determinant of $\mathbf{P}$ divides $u = \prod_{k=1}^{q} u_k$ ([3, Theorem 2.7(b)]). This is obvious initially, and with each incremental step the new basis matrix is derived from the old by multiplying by a product of elementary matrices, which multiplies the determinant by a constant or a constant multiple of $x - \beta_s$.

6 List decoding, soft-decision decoding

In this section we apply our general techniques to specific problems for error correcting codes. The basic theory can be found in a number of references such as [14, 17, 18]. Throughout the section we will be dealing with 2–variable poynomials only, so we change our notation slightly to fit with that established in the literature: henceforth n refers to the number of congruences in (1) and (2) (so replaces p).

When the number of errors in a received word exceeds half the minimum distance of the code there may be more than one codeword consistent with the error vector. As an alternative to trying to find a particular codeword, the decoder may attempt to generate a list of the consistent codewords, and then choose among these according to some criterion. This is known as *list decoding* (see Elias [6] and the references given there). A new technique for list decoding applied to Reed-Solomon codes was invented by Sudan [20]. The essential idea is that a received word is used to create a set of points and a 2-variable polynomial interpolating these points is sought.

In more detail, let F be a field and let $\{(x_k, y_k) \in F^2, k = 1, \ldots, n\}$ be a set of distinct points. Given integers d, t, Sudan's algorithm determines the set of polynomials $f(x)$ of degree at most d with $|\{k | f(x_k) = y_k\}| \geq t$. The first step is to derive a pair of integers u, v, whose definition need not concern us here. We then determine a non-zero polynomial $Q(x, y)$, whose $(1, d)$–degree is at most $u + dv$, such that $Q(x_k, y_k) = 0$ for all $k = 1, \ldots, n$. The set of factors $y - f(x)$ of Q with $f(x_k) = y_k$ for at least t values of k is determined, where t is derived from the parameters of the code, and from this the list of consistent codewords is obtained.

The conditions on Sudan's algorithm confine its applicability to low rate codes. Guruswami and Sudan [12] extended the algorithm to codes of all rates by requiring, in addition, that the polynomial $Q(x, y)$ have certain derivatives equal to zero, equivalently, have certain multiple zeros, at the interpolating points.

In another direction, Kötter and Vardy [13] employ a similar technique in the context of algebraic soft-decision decoding of Reed-Solomon codes. They derive a set of interpolation points and multiplicities from posterior probabilities. In their technique, the number of multiplicities is not, in general, the same at each point (whereas it is in the Guruswami and Sudan algorithm). Thus, we can see that both Sudan's original algorithm and the extension by Guruswami and Sudan are special cases of the method of Kötter and Vardy. In addition, Kötter and Vardy impose a condition of *minimality* of the required solution $Q(x, y)$ with respect to an (a, b)–degree for certain a, b. We shall show that our techniques may be applied to solve the Kötter-Vardy problem in a very natural way.

Roth and Ruckenstein [19] improved the efficiency of Sudan's algorithm by deriving an *extended key equation*, which they then solved using a modified form of the "Fundamental Iterative Algorithm" of Feng and Tzeng [7]. The main effect is to accelerate the interpolation step of the original algorithm. Suppose that the list of codewords sought is to contain at most s elements, then Sudan's algorithm can correct up to $\tau = \tau(s)$ errors. In [19], the received word is used to generate a set of polynomials $S^{(i)}$ and a solution $(\Lambda^{(1)}, \ldots, \Lambda^{(s)}, \Omega)$ is sought to the congruence

$$\sum_{i=1}^{s} \Lambda^{(i)} x^{(i-1)w} S^{(i)} \equiv \Omega(x) \bmod x^r \tag{13}$$

subject to the constraints $\partial\Lambda^{(i)} < n - \tau - iw, 1 \leq i \leq s, \partial\Omega(x) < r - \tau$, where w, r are derived from the parameters of the code. By setting $h_{s+1} = -1$ and $h_i =$

$x^{(i-1)w}S^{(i)}, 1 \leq t \leq s$, this can be recast as

$$\sum_{i=1}^{s+1} b_i h_i \equiv 0 \bmod x^r \tag{14}$$

subject to constraints of the form $\partial b_i \leq r_i, 1 \leq i \leq s+1$. We can apply the methods of Section 3 using the sequence of ideals $I_\ell = \langle x^\ell \rangle, 0 \leq \ell \leq r$ and the function θ_ℓ which returns the coefficient of x^ℓ in the expansion of the left hand side of (14). The term order chosen is an instance of that described in Section 4.

The most general version of the problem (as considered by Kötter and Vardy [13]) has the following form. A polynomial $Q(x, y)$ can be expanded around (x_k, y_k) to give $Q(x, y) = \sum_i \sum_j q_{ijk}(x - x_k)^i (y - y_k)^j$. It follows that $Q(x + x_k, y + y_k) = \sum_i \sum_j q_{ijk} x^i y^j$ for certain coefficients q_{ijk}. Given a set of points $(x_k, y_k), k = 1, \ldots, n$, and multiplicities m_k, we require the polynomial $Q(x, y)$, minimal with respect to some (a, b)–degree, such that

$$q_{ijk} = 0 \text{ for } 1 \leq k \leq n \text{ and } i + j < m_k.$$

In our terms this condition translates as

$$Q(x + x_k, y + y_k) \equiv 0 \bmod \langle \{x^i y^j | i + j = m_k\} \rangle \text{ for } k = 1, \ldots, n. \tag{15}$$

To transform this into a problem amenable to solution by our methods, we take (in the notation of Section 3) $A = F[x, y], q = 1, H^{(k)} : A \to A, H^{(k)}(Q(x, y)) = Q(x + x_k, y + y_k)$, so that $H^{(k)}(xQ(x, y)) = (x + x_k)Q(x + x_k, y + y_k) = (x + x_k)H^{(k)}(Q(x, y))$, giving $\gamma_1 = \gamma_x = x_k$, and similarly $\gamma_2 = \gamma_y = y_k$. We define $I^{(k)} = \langle \{x^i y^j | i + j = m_k\} \rangle$. As in [11], we may choose to work with terms in A in degree lexicographic order and define a descending sequence of $\frac{m_k(m_k+1)}{2}$ ideals $I_\ell^{(k)}$ between $A = \langle 1 \rangle$ and $I^{(k)}$ corresponding to terms $\varphi_\ell^{(k)}$, where $I_\ell^{(k)} = \langle \varphi_\ell^{(k)}, I_{\ell+1}^{(k)} \rangle$ and $xI_\ell^{(k)} \subseteq I_{\ell+1}^{(k)}, yI_\ell^{(k)} \subseteq I_{\ell+1}^{(k)}$ (giving condition (5) with $\beta_1 = \beta_x = 0, \beta_2 = \beta_y = 0$). The homomorphism $\theta_\ell^{(k)}$ returns the coefficient of $\varphi_\ell^{(k)}$ in the unique representative of an element of $I_\ell^{(k)}$ modulo $I_{\ell+1}^{(k)}$, thus satisfying (6). For the solution module, we use the term order defined by (a, b)–degree with ties broken by the degree of y (*cf.* Example 2.2). The specific polynomial sought in [13] is the minimal element of the solution module with respect to this term order, and, as such, can be identified as the first element in a strictly ordered Gröbner basis of the solution module. This polynomial requires the minimum amount of further processing in order to find the corresponding list of codewords.

More formally, the algorithm iterates, for each point (x_k, y_k), through the terms in degree-lexicographic order (where we choose $x < y$) as far as the last term with total degree less than m_k.

Algorithm 6.1

Input:

points $\{(x_k, y_k) : 1 \le k \le n\}$, with multiplicities m_k

Output:

Gröbner basis $\mathcal{W}$ of the solution module of (15)

the required Q as the first element of $\mathcal{W}$

Main Routine:

$\mathcal{W} := \{1\}$ (†)

For k from 1 to n

  For ℓ from 0 to $m_k - 1$

    For s from 0 to ℓ

      For j from 1 to $|\mathcal{W}|$

        α_j:=coefficient of $x^{\ell-s}y^s$ in $\mathcal{W}[j](x + x_k, y + y_k)$

      If $\alpha_j = 0$ for all j then

        $\mathcal{W}' = \mathcal{W}$

      else

        $j^* :=$ least j for which $\alpha_j \ne 0$

        $\mathcal{W}_1 := \{\mathcal{W}[j] : j < j^*\}$

        $\mathcal{W}_2 := \{(x - x_k)\mathcal{W}[j^*]\} \bigcup \{(y - y_k)\mathcal{W}[j^*]\}$ (‡)

        $\mathcal{W}_3 := \{\mathcal{W}[j] - (\alpha_j/\alpha_{j^*})\mathcal{W}[j^*] : j > j^*\}$

        $\mathcal{W}' := \mathcal{W}_1 \bigcup \mathcal{W}_2 \bigcup \mathcal{W}_3$

      $\mathcal{W}$:=ord($\mathcal{W}'$)

EXAMPLE 6.2 In this example we allow the function *ord* to construct ordered minimal Gröbner bases. With the $(2, 3)$–order defined in 2.2, let $n = 2, (x_1, y_1) = (1, 0), m_1 = 1, (x_2, y_2) = (0, 1), m_2 = 2$. Here we shall write $[f]_{x^u y^v}$ for the coefficient of $x^u y^v$ in f, and indicate only the α_j and $\mathcal{W}$ at each step. The iterations are as follows:

$\underline{(k, \ell, s) = (1, 0, 0)}$

$[1]_1 = 1 \to \{x - 1, y\}$

$\underline{(k, \ell, s) = (2, 0, 0)}$

$[x - 1]_1 = -1, [y + 1]_1 = 1 \to \{y + x - 1, x(x - 1)\}$ (here $\mathcal{W}_3 = \{y + x - 1\}$ and $(y - 1)(x - 1)$ has been eliminated in the formation of a minimal Gröbner basis)

$\underline{(k, \ell, s) = (2, 1, 0)}$

$[y + x]_x = 1, [x(x - 1)]_x = -1 \to \{x^2 + y - 1, x(y + x - 1), (y - 1)(y + x - 1)\}$

$\underline{(k, \ell, s) = (2, 1, 1)}$

$[x^2 + y]_y = 1, [x(y + x)]_y = 0, [y(y + x)]_y = 0 \to \{x(y + x - 1), x(x^2 + y - 1), (y - 1)(y + x - 1)\}$ □

This algorithm begins with a single element in the basis (step (†)), and the number of elements can change with each iteration at step (‡). In theory there is unlimited growth in the size of the basis as the computation proceeds. However, the interpolation problems treated in this section all impose upper bounds on the $(0, 1)$-degree of the interpolating polynomials, and on the degrees of terms required to have zero coefficients. We may incorporate these into a sufficiently large global bound, and hence derive algorithms in which the number of basis elements at each iteration is fixed. These will generally be more efficient for implementation purposes.

Suppose that each interpolating polynomial satisfying the weighted degree constraint and all relevant terms are contained in $B = \{f \in F[x, y] | \Delta_{(0,1)}(f) < q\}$ for some $q \in \mathbf{N}$. For convenience, we can identify B, as $F[x]$-module, with $F[x]^q$ using,

$$\psi : B \to F[x]^q,$$

$$\psi(x^i y^j) = x^i \mathbf{e}_{j+1}, 0 \leq j \leq q - 1,$$

and extending by linearity. The set $T = \{\mathbf{f} \in F[x]^q | \mathbf{f} = \psi(Q), Q \in B,\ Q$ an interpolating polynomial$\}$ is a submodule of $F[x]^q$. The (a, b)–degree term order defined above can be used in an obvious fashion in $F[x]^q$, where the degree of $x^i \mathbf{e}_{j+1}$ is $ai + b(j + 1)$ and ties are broken by $\mathbf{e}_1 < \mathbf{e}_2 < \mathbf{e}_3$. We can now use our algorithm to solve the 1–variable minimization problem, by finding a Gröbner basis of T.

Let $A = F[x]$, and define $H^{(k)} : A^q \to A^q$ by $H^{(k)}(\mathbf{b}) = \psi[\psi^{-1}(\mathbf{b})(x+x_k, y+y_k)]$. Now $\psi^{-1}(x\mathbf{b}) = x\psi^{-1}(\mathbf{b})$ and so $H^{(k)}(x\mathbf{b}) = \psi[(x + x_k)\psi^{-1}(\mathbf{b})(x + x_k, y + y_k)] = (x + x_k)\psi[\psi^{-1}(\mathbf{b})(x + x_k, y + y_k)] = (x + x_k)H^{(k)}(\mathbf{b})$, and $\gamma_1 = \gamma_x = x_k$. We define $M^{(k)} = \langle\{x^i \mathbf{e}_{j+1} | i + j = m_k\}\rangle$. Let $I_\ell^{(k)}$ denote the sequence of ideals defined in the preamble to Algorithm 6.1. Then $M_\ell^{(k)} = \psi(I_\ell^{(k)} \bigcap B)$ defines a corresponding descending sequence of $\frac{m_k(m_k+1)}{2}$ A–modules between A^q and $M^{(k)}$ and the terms $\psi(\varphi_\ell^{(k)})$ satisfy $M_\ell^{(k)} = \langle\psi(\varphi_\ell^{(k)}), M_{\ell+1}^{(k)}\rangle$, and $xM_\ell^{(k)} \subseteq M_{\ell+1}^{(k)}$ (giving condition (5) with $\beta_1 = \beta_x = 0$). The homomorphism $\theta_\ell^{(k)}$ returns the coefficient of $\psi(\varphi_\ell^{(k)})$. This results in a 1–variable algorithm with a Gröbner basis of size q at each step, in which for each component position there is precisely one element with leading term in that position. As noted in [16], in this situation the function *ord* takes a particularly simple form. The positions of the leading terms of the basis elements are unchanged by the incremental step, and thus re-ordering entails moving at most one element at each iteration.

Algorithm 6.3

Input:

points $\{(x_k, y_k) : 1 \le k \le n\}$, with multiplicities m_k

Output:

Gröbner basis $\mathcal{W}$ of the solution module

$\psi(Q)$ as the first element of $\mathcal{W}$

Main Routine:

$\mathcal{W} :=$ ord(standard basis of A^q) (†)

For k from 1 to n

For ℓ from 0 to $m_k - 1$

For s from 0 to ℓ

For j from 1 to q $(\star)$

α_j:=coefficient of $x^{\ell-s}\mathbf{e}_{s+1}$ in $\psi[\psi^{-1}(\mathcal{W}[j])(x + x_k, y + y_k)]$

If $\alpha_j = 0$ for all j then

$\mathcal{W}' := \mathcal{W}$

else

$j^* :=$ least j for which $\alpha_j \neq 0$

$\mathcal{W}_1 := \{\mathcal{W}[j] : j < j^*\}$

$\mathcal{W}_2 := \{(x - x_k)\mathcal{W}[j^*]\}$ (‡)

$\mathcal{W}_3 := \{\mathcal{W}[j] - (\alpha_j/\alpha_{j^*})\mathcal{W}[j^*] : j > j^*\}$

$\mathcal{W}' := \mathcal{W}_1 \bigcup \mathcal{W}_2 \bigcup \mathcal{W}_3$

$\mathcal{W}$:=ord($\mathcal{W}'$)

We illustrate this algorithm by reworking Example 6.2.

Example 6.4 We shall take $q = 3$. The initial basis is $\{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$. Again, the notation $[\,\cdot\,]_X$ will indicate the coefficient of the monomial X in the relevant module element. The iterations are as follows:

$\underline{(k, \ell, s) = (1, 0, 0)}$

$[\varphi(\varphi^{-1}(1, 0, 0)(x + 1, y)]_{1\mathbf{e}_1} = 1,$

$[\varphi(\varphi^{-1}(0, 1, 0)(x + 1, y)]_{1\mathbf{e}_1} = 0,$

$[\varphi(\varphi^{-1}(0, 0, 1)(x + 1, y)]_{1\mathbf{e}_1} = 0,$

$\to \{(x - 1, 0, 0), (0, 1, 0), (0, 0, 1)\}.$

Note that the leading terms have $(2, 3)$-degrees 2,3,6, respectively.

$\underline{(k, \ell, s) = (2, 0, 0)}$

$[\varphi(\varphi^{-1}(x - 1, 0, 0)(x, y + 1)]_{1\mathbf{e}_1} = -1,$

$[\varphi(\varphi^{-1}(0, 1, 0)(x, y + 1)]_{1\mathbf{e}_1} = 1,$

$[\varphi(\varphi^{-1}(0, 0, 1)(x, y + 1)]_{1\mathbf{e}_1} = 1,$

$\to \{(x - 1, 1, 0), (x(x - 1), 0, 0), (x - 1, 0, 1)\}.$

$\underline{(k, \ell, s) = (2, 1, 0)}$

$[\varphi(\varphi^{-1}(x - 1, 1, 0)(x, y + 1)]_{x\mathbf{e}_1} = 1,$

$[\varphi(\varphi^{-1}(x(x-1),0,0)(x,y+1)]_{x\mathbf{e}_1} = -1,$
$[\varphi(\varphi^{-1}(x-1,0,1)(x,y+1)]_{x\mathbf{e}_1} = 1,$
$\rightarrow \{(x^2-1,1,0),(x(x-1),x,0),(0,-1,1)\}.$
$\underline{(k,\ell,s) = (2,1,1)}$
$[\varphi(\varphi^{-1}(x^2-1,1,0)(x,y+1)]_{1\mathbf{e}_2} = 1,$
$[\varphi(\varphi^{-1}(x(x-1),x,0)(x,y+1)]_{1\mathbf{e}_2} = 0,$
$[\varphi(\varphi^{-1}(0,-1,1)(x,y+1)]_{1\mathbf{e}_2} = 1,$
$\rightarrow \{(x(x-1),x,0),(x(x^2-1),x,0),(-x^2+1,-2,1)\}.$
Note that the final basis corresponds to $\{x(y+x-1), x(x^2+y-1), y^2-x^2-2y+1\}$. The first two elements are the same as those in the basis derived in 6.2, while the third is obtained from that basis as $(y-1)(y+x-1) - x(y+x-1)$. □

We end with a variation of Algorithm 6.3 related to [15, Algorithm 15]. For any $\rho \in \mathbf{N}$, the set $\mathcal{W}_0 = \{x^i\mathbf{e}_j | 0 \le i \le \rho-1, 1 \le j \le q\}$ is a Gröbner basis of A^q with $|\mathcal{W}_0| = \rho q$ elements. With this as the initial basis, we obtain an algorithm with ρq elements in the basis at each iteration.

Algorithm 6.5	
Algorithm 6.3 with the following changes	
$\mathcal{W} := \mathrm{ord}(\mathcal{W}_0) = \mathrm{ord}(\{x^i\mathbf{e}_j \vert 0 \le i \le \rho-1, 1 \le j \le q\})$	(†)
for j from 1 to ρq	$(\star)$
$\mathcal{W}_2 := \{(x-x_k)^\rho \mathcal{W}[j^*]\}$	(‡)

Proof. We prove by induction that each iteration of the inner loop (the incremental step) produces a Gröbner basis $\{\mathcal{W}[1], \ldots, \mathcal{W}[\rho q]\}$ for next solution submodule in the sequence and that the sets $P_i = \{x^{m\rho}\mathrm{lt}(\mathcal{W}[i]) | m \in \mathbf{N}_0\}, i = 1, \ldots, \rho q$ partition the leading terms of that submodule. The assertion is obviously true for the initial basis $\mathcal{W}_0$. Let us now fix k, ℓ and drop the superscripts (k). As in Theorem 3.1, let $S \subseteq A^q$ be a submodule such that $H(\mathbf{b}) \equiv 0 \bmod M_\ell$ for all $\mathbf{b} \in S$, and $S' = \{\mathbf{b} \in A^q | H(\mathbf{b}) \equiv 0 \bmod M_{\ell+1}\}$.

We assume that $\mathcal{W}$ is a Gröbner basis for S and that the $P_i = \{x^{m\rho}\mathrm{lt}(\mathcal{W}[i]) | m \in \mathbf{N}_0\}, i = 1, \ldots, \rho q$ partition the leading terms of S. Since $(x-x_k)\mathcal{W}[j^*] \in S'$ it follows that $(x-x_k)^\rho\mathcal{W}[j^*]$ is in S'. Also, $\mathrm{lt}((x-x_k)^\rho\mathcal{W}[j^*]) = x^\rho\mathrm{lt}(\mathcal{W}[j^*])$. Following the proof of Theorem 3.1, we may assume $S' \subseteq S$, and observe that $\mathrm{lt}(\mathcal{W}[j^*])$ is the only leading term that changes at this iteration. We may therefore restrict our attention to those $\mathbf{f} \in S'$ such that $\mathrm{lt}(\mathcal{W}[j^*])$ divides $\mathrm{lt}(\mathbf{f})$.

By definition, $\mathrm{lt}(\mathbf{f})$ is in one and only one subset P_i of the partition. If $i \ne j^*$ then $\mathrm{lt}(\mathbf{f})$ is divisible by the leading term of a basis element other than $\mathcal{W}[j^*]$. We can therefore assume $\mathrm{lt}(\mathbf{f}) \in P_{j^*}$ and thus $\mathrm{lt}(\mathbf{f}) = x^{m_f\rho}\mathrm{lt}(\mathcal{W}[j^*])$ for some $m_f \in \mathbf{N}_0$. If $\mathrm{lt}(\mathbf{f}) \ne \mathrm{lt}(\mathcal{W}[j^*])$ then $\mathrm{lt}(\mathbf{f})$ is a multiple of $x^\rho\mathrm{lt}(\mathcal{W}[j^*])$, which is what we want.

Otherwise, $\mathrm{lt}(\mathbf{f}) = \mathrm{lt}(\mathcal{W}[j^*])$. Let $\mathbf{f}^* = \mathbf{f} - (\mathrm{lc}(\mathbf{f})/\mathrm{lc}(\mathcal{W}[j^*]))\mathcal{W}[j^*]$ and note that $\mathbf{f}^* \in S$. Now, $\mathrm{lt}(\mathbf{f}^*) < \mathrm{lt}(\mathcal{W}[j^*])$ and

$$\begin{aligned}
\theta_\ell(H(\mathbf{f}^*)) &= \theta_\ell(H(\mathbf{f})) - (\mathrm{lc}(\mathbf{f})/\mathrm{lc}(\mathcal{W}[j^*]))\theta_\ell(H(\mathcal{W}[j^*])) \\
&= -(\mathrm{lc}(\mathbf{f})/\mathrm{lc}(\mathcal{W}[j^*]))\theta_\ell(H(\mathcal{W}[j^*])) \\
&\neq 0
\end{aligned}$$

so $\mathbf{f}^*$ is not an element of S'. However, the standard representation of $\mathbf{f}^*$ contains only basis elements whose leading terms are less than $\mathrm{lt}(\mathcal{W}[j^*])$ and which are therefore elements of S'. This is a contradiction, and we conclude that the new basis is a Gröbner basis of S'.

Next, let $P'_i = P_i$ for $i \neq j^*$ and $P'_{j^*} = \{x^{m\rho}(x^\rho \mathrm{lt}(\mathcal{W}[j^*]))|m \in \mathbf{N}_0\}$. Since $P'_{j^*} \subset P_{j^*}$, the sets $P'_i, i = 1, \ldots, \rho q$ are disjoint. Thus the new basis has no duplicate leading terms and the function *ord* does not remove any basis elements. Suppose that $\mathbf{f} \in S'$ and $\mathrm{lt}(\mathbf{f}) \notin \bigcup_{i \neq j^*} P'_i (= \bigcup_{i \neq j^*} P_i)$. From the previous paragraph, $\mathrm{lt}(\mathbf{f}) = x^{m_f \rho}\mathrm{lt}(\mathcal{W}[j^*])$ for some $m_f \in \mathbf{N}_0, m_f \neq 0$. It now follows that $\bigcup_{i=1}^{\rho q} P'_i$ partitions the set of leading terms of S'. $\square$

Example 6.6 Once again, we reformulate Example 6.2. Verification that the final basis is a Gröbner basis of the same submodule as that obtained in 6.2 and 6.4 is left as an exercise. Here $\rho = 3$ and the sequence of bases is as follows:
Initially $\{(1,0,0),(x,0,0),(0,1,0),(x^2,0,0),(0,x,0),(0,0,1),$
$(0,x^2,0),(0,0,x),(0,0,x^2)\}$
$(k,l,s) = (1,0,0)$
$\{(x-1,0,0),(0,1,0),(x^2-1,0,0),(0,x,0),((x-1)^3,0,0),$
$(0,0,1),(0,x^2,0),(0,0,x),(0,0,x^2)\}$
$(k,l,s) = (2,0,0)$
$\{(x-1,1,0),(x(x-1),0,0),(0,x,0),((x^3-3x^2+2x,0,0),$
$(x-1,0,1),(0,x^2,0),(0,0,x),(x^3(x-1),0,0),(0,0,x^2)\}$
$(k,l,s) = (2,1,0)$
$\{(x^2-1,1,0),(-(x-1),x-1,0),(x^3-3x^2-2,-2,0),(0,-1,1),(0,x^2,0),$
$(x^3(x-1),0,0),(-(x-1),1,,x),(x^3(x-1),x^3,0),(0,0,x^2)\}$
$(k,l,s) = (2,1,1)$
$\{(x(x-1),x,0),(x^2(x-1),0,0),(-x^2+1,-2,1),(0,x^2,0),(x^3(x-1),0,0),$
$(x^2-x,0,x),(x^3(x-1),x^3,0),(x^3(x^2-1),x^3,0),(0,0,x^2)\}$ $\square$

For the Guruswami and Sudan formulation, Algorithm 6.5 coincides with the technique given by Nielsen and Hoeholdt [15, Algorithm 15]. The term order used there is based on $(1, K-1)$–degree, where K is the dimension of the relevant Reed-Solomon code. For a multiplicity ρ the initial basis is $\{x^i y^j | 0 \leq i \leq \rho - 1, 0 \leq j \leq N\}$, where N is the maximum $(0, 1)$–degree required, which corresponds to the basis used above.

References

[1] W.W. Adams, P. Loustaunau, *An Introduction to Gröbner bases*, Springer-Verlag, New York, Berlin, 1994.

[2] T. Becker, V. Weispfenning, *Gröbner Bases: A Computational Approach to Commutative Algebra*, Springer Verlag, 1993.

[3] B. Beckermann, G. Labahn, Recursiveness in matrix rational interpolation problems , *J. of Computational and Applied Mathemtics*, 77 (1997), 5–34

[4] D. Cox, J. Little, D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Springer-Verlag, New York, Berlin, 1992.

[5] D. Eisenbud, *Commutative Algebra with a View towards Algebraic Geometry*, Graduate Texts in Mathematics, Springer-Verlag, 1995.

[6] P. Elias, Error-correcting codes for list decoding, *IEEE Trans. on Inform. Theory*, IT–37 (1991), 5–12.

[7] G.L. Feng, K.K. Tzeng, A generalization of the Berlekamp-Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes, *IEEE Trans. Inform. Theory*, 37 (1991), 1274–1287.

[8] P. Fitzpatrick. On the key equation, *IEEE Trans. on Inform. Theory*, IT–41 (1995), 1290–1302.

[9] P. Fitzpatrick. On the scalar rational interpolation problem, *Mathematics of Control, Signals, and Systems*, 9 (1996), 352–369.

[10] P. Fitzpatrick. Rational interpolation using Gröbner bases: some numerical results, in *Mathematics in Signal Processing* IV, J.G. McWhirter, ed., Clarendon Press, Oxford, 1997, 35–45.

[11] P. Fitzpatrick. Solving a multivariable congruence by change of term order, *J. Symb. Comp.*, 11 (1997), 505–510.

[12] V. Guruswami, M. Sudan, Improved decoding of Reed-Solomon and algebraic-geometry codes, *IEEE Trans. on Inform. Theory*, IT–45 (1999), 1757–1767.

[13] R. Kötter, A. Vardy, Algebraic soft-decision decoding of Reed-Solomon codes, presented at *ISIT* 2000, Sorrento, June 2000.

[14] F.J. McWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.

[15] R.R. Nielsen,T. Hoeholdt, Decoding Reed-Solomon codes beyond half the minimum distance, *Proceedings of an International Conference on Coding Theory, Cryptography and Related Areas*, Guanajuato, April 1998, Springer-Verlag, 2000.

[16] H. O'Keeffe, P. Fitzpatrick, Recursive construction of Gröbner bases for the solution of polynomial congruences, *Codes, Systems and Graphical models, Vol 123 IMA Volumes in Mathematics and its Applications*, 2000, 299–311.

[17] V.S. Pless, W.C. Huffman (eds), *Handbook of Coding Theory*, Elsevier, Amsterdam, 1998.

[18] W.W. Peterson, E.J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed., MIT, Cambridge, MA, 1972.

[19] R. Roth, G. Ruckenstein, Efficient decoding of Reed-Solomon codes beyond half the minimum distance, *IEEE Trans. on Inform. Theory* IT–46 (2000), 246–257

[20] M. Sudan, Decoding of Reed-Solomon codes beyond the error correction bound, *J. of Complexity*, 13 (1997), 180–193.