GRÖBNER BASES, PADÉ APPROXIMATION, AND DECODING OF LINEAR CODES

JEFFREY B. FARR AND SHUHONG GAO

Abstract. This paper shows how Gröbner basis techniques can be used in coding theory, especially in the construction and decoding of linear codes. A simple algorithm is given for computing the reduced Gröbner basis of the vanishing ideal of a given set of finitely many points, and it is used for finding Padé approximation of any polynomial (given implicitly), which is a major step in decoding. A new method is given for construction of a large class of linear codes that can also be decoded efficiently. These codes include as special cases many of the well known codes such as Reed-Solomon codes, Hermitian codes and, more generally, all one-point algebraic geometry codes.

1. Introduction

Let q be a prime power and $\mathbb{F}_q$ denote the finite field with q elements. A general framework for constructing linear codes over $\mathbb{F}_q$ is as follows. Suppose $V = \{P_1, \ldots, P_n\}$ is a set of n distinct points in $\mathbb{F}_q^m$ and L a vector space over $\mathbb{F}_q$ of functions on V with values in $\mathbb{F}_q$. Thus, $f(P_i) \in \mathbb{F}_q$ for all i and for $f \in L$. Define

$$\mathcal{C} = \{(f(P_1), \ldots, f(P_n)) : f \in L\};$$

that is, $\mathcal{C}$ is the image of L under evaluation at the points in V. Then $\mathcal{C}$ is a linear subspace in $\mathbb{F}_q^n$, thus a linear code of length n over $\mathbb{F}_q$. It is well known that every linear code can be obtained this way.

To obtain useful codes, one has to choose the point set V and the function space L carefully. A powerful method is to use algebraic geometry, namely, to require the points to lie on a certain curve (or an algebraic variety) and the functions in L to have certain pole orders. Results from algebraic geometry enable one to get bounds on the minimum distance of the codes constructed. Many families of good codes have been constructed this way. This approach requires significant study in algebraic geometry. For more details, the reader is referred to the excellent surveys [6, 17] and textbooks [16, 25]. There is an effort to introduce an elementary approach that is more accessible to engineers. The idea is to still use points on curves but to introduce order and weight functions in place of pole orders in the algebraic geometry setting. Here one needs to find an appropriate weight function

Date: August 6, 2004.

This work was supported in part by National Science Foundation (NSF) under Grant DMS0302549, National Security Agency (NSA) under Grant MDA904-02-1-0067, the DoD Multidisciplinary University Research Initiative (MURI) program administered by the Office of Naval Research (ONR) under Grant N00014-00-1-0565.

(which may not exist at all in certain cases). Work in this direction is well described in the survey paper [17].

In this paper, we describe our recent effort to get a simpler approach for construction and decoding of linear codes. We use monomial orders (corresponding roughly to the order functions mentioned above) and Gröbner basis theory. The novelty in our construction is that we have a natural decoding algorithm. Our construction includes codes from algebraic geometry as special cases. We can also construct a large class of random codes, and for them our decoding algorithm performs well compared with Shannon's entropy bound.

The rest of the paper is organized as follows. In Section 2, we give a brief introduction to Gröbner bases with the basic notations and results. This is intended for readers who are not familiar with this area. In Section 3, we present a simple algorithm for computing Gröbner bases for a special class of ideals, and in the following section we show how our algorithm can be used in multivariate Padé approximation, which is a major step in our decoding algorithm. In Section 5, we present our construction of linear codes using Gröbner basis theory, and we give a decoding algorithm for these codes. In the final section, we make a few comments on our method and some further research problems.

2. Gröbner Bases

Let $\mathbb{F}$ be any field, and let $\mathbf{R} = \mathbb{F}[x_1, \ldots, x_m]$ denote the polynomial ring in m variables $x_1, \ldots, x_m$. A nonempty subset $\mathbf{I} \subseteq \mathbf{R}$ is called an *ideal* if

(i) $a, b \in \mathbf{I} \Rightarrow a + b \in \mathbf{I}$, and
(ii) $a \in \mathbf{I} \Rightarrow ab \in \mathbf{I}$ for every $b \in \mathbf{R}$.

Any collection of polynomials $f_1, \ldots, f_n \in \mathbf{R}$ generates an ideal of $\mathbf{R}$ in a natural way, namely, the set of all polynomials of the form

$$h_1 f_1 + \cdots + h_n f_n,$$

where $h_i \in \mathbf{R}$ are arbitrary. This ideal is denoted as $\langle f_1, \ldots, f_n \rangle$. For an ideal $\mathbf{I}$ of $\mathbf{R}$, if $\mathbf{I} = \langle f_1, \ldots, f_n \rangle$ then we say that $\mathbf{I}$ is generated by $f_1, \ldots, f_n$, or $\{f_1, \ldots, f_n\}$ is a *basis* for $\mathbf{I}$. An ideal may have many bases, and different bases may have different number of elements. Also, the elements of a basis for an ideal need not be linearly independent over $\mathbb{F}$ but can be easily reduced to linearly independent ones. Hilbert's basis theorem guarantees that every ideal $\mathbf{I}$ in $\mathbf{R}$ can be generated by finitely many polynomials.

The ideal $\mathbf{I} = \langle f_1, \ldots, f_n \rangle$ captures the common zeros of $f_1, \ldots, f_n$ nicely in the sense that

1. every common zero of $f_1, \ldots, f_n$ is a zero of every $g \in \mathbf{I}$; and
2. if $\mathbf{I}$ has another basis, say $\mathbf{I} = \langle g_1, \ldots, g_s \rangle$, then a point in $\mathbb{F}^m$ is a common zero of $f_1, \ldots, f_n$ if and only if it is a common zero of $g_1, \ldots, g_s$.

A *Gröbner basis* for an ideal $\mathbf{I} \subseteq \mathbf{R}$ is some "nice" basis for $\mathbf{I}$ that enables us to better "control" the common zeros of the ideal and to perform computation related to the ideal (*e.g.* testing whether a given polynomial $g \in \mathbf{R}$ belongs to $\mathbf{I}$). As a simple example, consider two polynomials $f_1, f_2 \in \mathbb{F}[x]$ (univariate polynomials) and the ideal $\mathbf{I} = \langle f_1(x), f_2(x) \rangle$ in $\mathbb{F}[x]$. Let $d(x) = \gcd(f_1(x), f_2(x))$. Then we have $\mathbf{I} = \langle d(x) \rangle$. So $d(x)$ itself is a basis for the ideal, and it is nice in the sense that the degree of $d(x)$ is equal to the number of common zeros of $f_1(x)$ and $f_2(x)$ (in the algebraic closure of $\mathbb{F}$) and a polynomial $g(x) \in \mathbb{F}[x]$ belongs to $\mathbf{I}$ if and only if $g(x)$ is divisible by $d(x)$. For multivariate polynomials, a Gröbner basis for an ideal provides similar information and much more. To properly define a Gröbner basis, we first need to introduce orders among monomials.

2.1. **Monomial orders.** We use the following notations. Let $\mathbb{N} = \{0, 1, 2, \dots\}$, $\mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}$, and $\mathbb{R}$ be the set of real numbers. For $\alpha = (\alpha_1, \dots, \alpha_m) \in \mathbb{N}^m$,

$$\mathbf{x}^\alpha = x_1^{\alpha_1} \dots x_m^{\alpha_m},$$

is called a *monomial* in $\mathbb{F}[x_1, \dots, x_m]$. Hence, there is a one-to-one correspondence between monomials in $\mathbb{F}[x_1, \dots, x_m]$ and elements in $\mathbb{N}^m$.

A *monomial order* on $\mathbb{F}[x_1, \dots, x_m]$ is any total ordering $>$ on all the monomials such that

1. $\mathbf{x}^\alpha > 1$ for every $\alpha \in \mathbb{N}^m$ with $\alpha \neq \mathbf{0}$;
2. If $\mathbf{x}^\alpha > \mathbf{x}^\beta$, then $\mathbf{x}^\alpha \cdot \mathbf{x}^\gamma > \mathbf{x}^\beta \cdot \mathbf{x}^\gamma$ for all $\gamma \in \mathbb{N}^m$.

By "total ordering" above, we mean that any two monomials should be comparable; that is, they are either equal or one is bigger than the other. The second condition implies that if a polynomial is multiplied by a monomial, the ordering of its terms will not change. This property is important for long division of polynomials. Also, the two conditions imply that if $\mathbf{x}^\alpha$ is divisible by $\mathbf{x}^\beta$ then $\mathbf{x}^\alpha \geq \mathbf{x}^\beta$.

A natural way to define a monomial order is to use weighted degrees. Let $\mathbf{w} = (w_1, \dots, w_m) \in \mathbb{R}^m$ be a vector of real numbers. For any monomial $\mathbf{x}^\alpha = x_1^{\alpha_1} \dots x_m^{\alpha_m}$, its $\mathbf{w}$-*weighted degree*, or simply $\mathbf{w}$-*degree*, is defined to be

$$\mathbf{w} \cdot \alpha = w_1 \alpha_1 + \cdots + w_m \alpha_m.$$

When $\mathbf{w} = (1, \cdots, 1)$, then the $\mathbf{w}$-degree is the same as the the total degree. When $\mathbf{w}$ is the i-th unit vector (which has all coordinates zero but the i-th being 1), then the $\mathbf{w}$-degree is equal to the degree in x_i. We use a sequence of weight vectors $\mathbf{w}_1, \dots, \mathbf{w}_\ell \in \mathbb{R}^m$ to define a monomial order as follows. We say that $\mathbf{x}^\alpha > \mathbf{x}^\beta$ if for some i $(1 \leq i \leq \ell)$,

$$\mathbf{w}_j \cdot \alpha = \mathbf{w}_j \cdot \beta, \quad 1 \leq j \leq i-1, \qquad \text{but } \mathbf{w}_i \cdot \alpha > \mathbf{w}_i \cdot \beta.$$

This means that a monomial is bigger if it has higher $\mathbf{w}$-degree: using first $\mathbf{w}_1$-degree, then $\mathbf{w}_2$-degree (to break tie), then $\mathbf{w}_3$-degree, ..., and finally $\mathbf{w}_\ell$-degree. The weight vectors $\mathbf{w}_1, \dots, \mathbf{w}_\ell$ are usually presented as an $\ell \times m$ matrix whose ith row is $\mathbf{w}_i$, $1 \leq i \leq \ell$. One can check that the matrix W defines a monomial order if and only if the following conditions are satisfied:

(a) there is no nonzero $\alpha \in \mathbb{Z}^m$ such that $\mathbf{w}_i \cdot \alpha = 0$ for all $1 \leq i \leq \ell$, and
(b) the first nonzero entry of each column of W is positive.

For example, we can define a monomial order on $\mathbb{F}[x_1, x_2, x_3]$ (or $\mathbb{F}[x, y, z]$) using any one of the following matrices:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & \pi & \pi^2 \end{pmatrix},$$

where $\pi = 3.1415926\cdots$. The first matrix defines the *lexicographical order* with $x_1 > x_2 > x_3$, and the second the *graded lex order* with $x_1 > x_2 > x_3$. It is a challenge for the reader to prove that the third matrix satisfies the condition (a) above.

2.2. **Gröbner bases.** Fix any monomial order $>$ on $\mathbb{F}[x_1, \ldots, x_m]$. For any nonzero polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$, its terms can be arranged in decreasing order:

$$f = a\mathbf{x}^\alpha + \sum_{\beta<\alpha} a_\beta \mathbf{x}^\beta,$$

where $a, a_\beta \in \mathbb{F}$ are nonzero. The first term $a\mathbf{x}^\alpha$ is called the *leading term* of f, denoted by $\mathrm{LT}(f)$; $\mathbf{x}^\alpha$ is called the *leading monomial* of f; denoted by $\mathrm{LM}(f)$, and the coefficient a of the leading term is called the *leading coefficient* of f, denoted by $\mathrm{LC}(f)$. Also, if $\mathbf{I}$ is a set of polynomials then

$$\mathrm{LT}(\mathbf{I}) = \{\mathrm{LT}(f) : f \in \mathbf{I}\}.$$

Let $\mathbf{I}$ be any ideal in $\mathbb{F}[x_1, \ldots, x_m]$, and fix any monomial order. A set of polynomials $g_1, \ldots, g_s \in \mathbf{I}$ is called a *Gröbner basis* if $\langle \mathrm{LT}(\mathbf{I}) \rangle = \langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s) \rangle$. That is, the leading term of every nonzero polynomial in $\mathbf{I}$ is divisible by some $\mathrm{LT}(g_i)$, $1 \le i \le s$. Intuitively, a Gröbner basis of $\mathbf{I}$ contains the smallest polynomials in $\mathbf{I}$ under a given monomial order. By this definition, a Gröbner basis remains a Gröbner basis if more polynomials are added to it, so an ideal may have many Gröbner bases. One can prove that an ideal in $\mathbb{F}[x_1, \ldots, x_m]$ always has a Gröbner basis.

For any $f \in \mathbb{F}[x_1, \ldots, x_m]$, we may perform long division on f using the polynomials in the Gröbner basis G as divisors. This process is called "reducing f by G," and the resulting fully-reduced polynomial is said to be in normal form, denoted $Normal(f, G)$. We call a Gröbner basis *reduced* if every polynomial in the basis is reduced with respect to each other. The reduced Gröbner basis for an ideal with respect to a given monomial order is unique and will be the focus of our attention in later sections.

2.3. **Quotient rings and monomial bases.** Let $\mathbf{I}$ be any ideal in $\mathbb{F}[x_1, \ldots, x_m]$. For $f, g \in \mathbb{F}[x_1, \ldots, x_m]$, we say that $f \equiv g \pmod{\mathbf{I}}$ if $f - g \in \mathbf{I}$. For example, if $f \in \mathbf{I}$, then $f \equiv 0 \pmod{\mathbf{I}}$. The following properties on congruence are easy to verify.

If $f_1 \equiv g_1 \pmod{\mathbf{I}}$ and $f_2 \equiv g_2 \pmod{\mathbf{I}}$, then
(1) $f_1 + f_2 \equiv g_1 + g_2 \pmod{\mathbf{I}}$
(2) $f_1 \cdot f_2 \equiv g_1 \cdot g_2 \pmod{\mathbf{I}}$
(3) $h \cdot f_1 \equiv h \cdot g_1 \pmod{\mathbf{I}}$ for any $h \in \mathbb{F}[x_1, \ldots, x_m]$.

Note that the converse of property (3) is not true, that is, if $h \cdot f \equiv h \cdot g \pmod{\mathbf{I}}$, one may not have $f \equiv g \pmod{\mathbf{I}}$.

The congruence relation above is an equivalence relation, so $\mathbb{F}[x_1, \ldots, x_m]$ is partitioned into equivalence classes, called *congruence classes* modulo $\mathbf{I}$. Each $f \in \mathbb{F}[x_1, \ldots, x_m]$ is in a unique congruence class, namely $f + \mathbf{I} = \{f + h : h \in \mathbf{I}\}$, often denoted by $\bar{f}$ or $[f]$. For $f, g \in \mathbb{F}[x_1, \ldots, x_m]$, we have $f + \mathbf{I} = g + \mathbf{I}$ (as sets) if and only if $f \equiv g \pmod{\mathbf{I}}$. The collection of all congruence classes is denoted by $\mathbb{F}[x_1, \ldots, x_m]/\mathbf{I}$.

We can make $\mathbb{F}[x_1, \ldots, x_m]/\mathbf{I}$ into a ring by defining the two operations:

$$[f] + [g] = [f + g], \quad [f] \cdot [g] = [fg],$$

which are simply polynomial addition and multiplication modulo $\mathbf{I}$. The properties (1) and (2) above imply that the two operations are well-defined and make $\mathbb{F}[x_1, \ldots, x_m]/\mathbf{I}$ into a *ring*. This ring is commutative and contains the field $\mathbb{F}$. In the following, we shall omit the brackets; that is, we view the elements in $\mathbb{F}[x_1, \ldots, x_m]/\mathbf{I}$ as polynomials in $\mathbb{F}[x_1, \ldots, x_m]$ but with two polynomials f and g identified whenever $f \equiv g \pmod{\mathbf{I}}$.

Fix any monomial order on $\mathbb{F}[x_1, \ldots, x_m]$. For any set $G \subset \mathbb{F}[x_1, \ldots, x_m]$, define

$$\mathcal{B}(G) = \{\mathbf{x}^\alpha : \alpha \in \mathbb{N}^m \text{ and } \mathbf{x}^\alpha \text{ not divisible by any } \mathrm{LT}(g), g \in G\}.$$

Suppose an ideal $\mathbf{I}$ is generated by the polynomials in G, *i.e.*, $\mathbf{I} = \langle G \rangle$. Then $\mathcal{B}(\mathbf{I}) \subseteq \mathcal{B}(G)$, and the equality holds if and only if G is a Gröbner basis. The monomials in $\mathcal{B}(\mathbf{I})$ have a special property: for any finite number of distinct monomials $\mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_t} \in \mathcal{B}(\mathbf{I})$, there are no elements $c_1, \ldots, c_t \in \mathbb{F}$, not all zero, such that

$$c_1\mathbf{x}^{\alpha_1} + \cdots + c_t\mathbf{x}^{\alpha_t} \equiv 0 \pmod{\mathbf{I}}.$$

That is, the monomials in $\mathcal{B}(\mathbf{I})$ are linearly independent modulo $\mathbf{I}$. In fact, $\mathcal{B}(\mathbf{I})$ is a basis (in the sense of linear algebra) for $\mathbb{F}[x_1, \ldots, x_m]/\mathbf{I}$ as a vector space over $\mathbb{F}$. We call $\mathcal{B}(\mathbf{I})$ the *monomial basis* for $\mathbb{F}[x_1, \ldots, x_m]/\mathbf{I}$ under the given monomial order. In the following, we simply say that $\mathcal{B}(\mathbf{I})$ is a monomial basis for $\mathbf{I}$. One can prove that $\mathcal{B}(\mathbf{I})$ is finite if and only if the polynomials in $\mathbf{I}$ have only finitely many common zeros (over the algebraic closure of $\mathbb{F}$). In the case that $\mathcal{B}(\mathbf{I})$ is finite, we say that $\mathbf{I}$ is *zero-dimensional*. Note that if the monomial order changes, the corresponding monomial basis may vary as well. These properties of $\mathcal{B}(\mathbf{I})$ are very important, and they will be used later in the construction of codes.

The above is an extremely brief introduction to the basics of Gröbner bases and their applications. There are many other issues that we cannot even touch upon here. The reader is referred to [2, 8, 19] for more details.

3. Vanishing Ideals of Distinct Points

We now consider a special class of ideals. Again, let $\mathbb{F}$ be any field. For any subset $V \subseteq \mathbb{F}^m$, define

$$\mathbf{I}(V) = \{f \in \mathbb{F}[x_1, \ldots, x_m] : f(P) = 0, \text{ for all } P \in V\},$$

that is, the set of polynomials that vanish on all the points in V. One can easily check that $\mathbf{I}(V)$ is an ideal in $\mathbb{F}[x_1, \ldots, x_m]$; it is called the *vanishing ideal* of V. If $V = \{P_1, \ldots, P_n\}$, $\mathbf{I}(V)$ is also written as $\mathbf{I}(P_1, \ldots, P_n)$. In coding theory and other applications (see, for example, [24] for statistics and [23] for biology), the points are given, and one is interested in polynomials that vanish on the points and are "smallest" in a certain sense. Gröbner basis methods provide an efficient tool for this purpose.

More precisely, the problem we consider is, given any distinct points $P_1, \ldots, P_n \in \mathbb{F}^m$ and any monomial order on $\mathbb{F}[x_1, \ldots, x_m]$, to compute the reduced Gröbner basis for the vanishing ideal $\mathbf{I}(P_1, \ldots, P_n)$. A polynomial time algorithm for this problem was first given by Buchberger and Möller (1982) [7] and significantly improved by Marinari, Möller and Mora (1993) [21] and Abbott, Bigatti, Kreuzer and Robbiano (2000) [1]. These algorithms perform Gauss elimination on a generalized Vandermonde matrix and have a polynomial time complexity. Recently, O'Keeffe and Fitzpatrick (2002) [22] studied this problem from a coding theory point of view. They present an algorithm that is exponential in the number of variables, and the Gröbner basis which they compute is not reduced.

We present an alternate method that is a generalization of Newton's interpolation for univariate polynomials. Our algorithm is similar to the approach of O'Keeffe and Fitzpatrick but computes the reduced Gröbner basis. Even though the time complexity of our algorithm is still exponential, its practical performance improves upon both O'Keeffe and Fitzpatrick's algorithm and the linear algebra approach mentioned above when the number of variables is relatively small compared to the number of points. We provide running time comparisons based on computer experiments for various monomial orders. We also present a preprocessing technique that significantly enhances the performance of our algorithm, the O'Keeffe-Fitzpatrick algorithm and, surprisingly, even the Gauss elimination algorithms.

3.1. **Algorithm.** In this section we present a solution to the problem of computing a Gröbner basis for the vanishing ideal of a finite set of distinct points. Throughout this section, we fix an arbitrary monomial order on $\mathbb{F}[x_1, \ldots, x_m]$, and $P_1, \ldots, P_n$ are n distinct points in $\mathbb{F}^m$.

We first state a result that gives a simple criterion on when a set of polynomials form a Gröbner basis for $\mathbf{I} = \mathbf{I}(P_1, \ldots, P_n)$.

Lemma 1. *For $g_1, \ldots, g_s \in \mathbf{I} = \mathbf{I}(P_1, \ldots, P_n)$, $\{g_1, \ldots, g_s\}$ is a Gröbner basis for $\mathbf{I}$ if and only if $|\mathcal{B}(g_1, \ldots, g_s)| = n$.*

The proof is omitted, as it follows from standard results in the literature. Our algorithm is based on the following lemma.

Lemma 2. *Suppose $G = \{g_1, \ldots, g_s\}$ is a Gröbner basis for $\mathbf{I}(V)$, for a finite set $V \subset \mathbb{F}^m$. For a point $P = (a_1, \ldots, a_m) \notin V$, let g_i denote the polynomial in G with smallest leading term such that $g_i(P) \neq 0$, and define*

$$\begin{aligned} \tilde{g}_j &:= g_j - \frac{g_j(P)}{g_i(P)} \cdot g_i, \qquad j \neq i, \ \textit{and} \\ g_{ik} &:= (x_k - a_k) \cdot g_i, \qquad 1 \leq k \leq m. \end{aligned}$$

Then

$$\widetilde{G} = \{\tilde{g}_1, \ldots, \tilde{g}_{i-1}, \tilde{g}_{i+1}, \ldots, \tilde{g}_s, g_{i1}, \ldots, g_{im}\}$$

is a Gröbner basis for $\mathbf{I}(V \cup \{P\})$.

Proof: Since $P \notin V$, at least one polynomial in G must be nonzero when evaluated at P; hence, g_i exists.

Certainly, $\widetilde{G} \subseteq \mathbf{I}(V \cup \{P\})$ as the new and modified polynomials evaluate to zero at all points in $V \cup \{P\}$. Denote $\mathrm{LT}(g_i)$ by $\mathbf{x}^\alpha$. We claim that

$$\mathcal{B}(\widetilde{G}) = \mathcal{B}(G) \cup \{\mathbf{x}^\alpha\}. \tag{1}$$

By the choice of i, $\mathrm{LT}(\tilde{g}_j) = \mathrm{LT}(g_j)$, for all $j \neq i$. Also, since g_i was replaced in $\widetilde{G}$ by $g_{i1}, g_{i2}, \ldots, g_{im}$, whose leading terms are $\mathbf{x}^\alpha x_1$, $\mathbf{x}^\alpha x_2$, $\ldots$, $\mathbf{x}^\alpha x_m$, we know that $\mathbf{x}^\alpha$ is the only monomial not in $\mathcal{B}(\mathbf{I}(V))$ that is in $\mathcal{B}(\mathbf{I}(V \cup \{P\}))$. Thus, (1) is satisfied, and $|\mathcal{B}(\widetilde{G})| = |\mathcal{B}(G)| + 1$. Since G is a Gröbner basis for $\mathbf{I}(V)$, we have $|\mathcal{B}(G)| = |V|$, and the conclusion follows from Lemma 1. □

Notice that for some of the g_{ik}, $\mathrm{LT}(g_{ik})$ may be divisible by the leading term of another polynomial in $\widetilde{G}$. In such a case, g_{ik} may be omitted from $\widetilde{G}$. In fact, we can check for this property before computing g_{ik} so that we save ourselves needless computation. In so doing, we also guarantee that the resulting $\widetilde{G}$ is a minimal Gröbner basis for $\mathbf{I}(V \cup \{P\})$.

We must, however, be even more careful if we wish to compute the (unique) reduced Gröbner basis. Notice that the reduction of any polynomial $g \in G$ with respect to $G \backslash \{g\}$ requires the use of only those polynomials in G which have leading term smaller than $\mathrm{LT}(g)$. Thus, it is easily seen that the $\tilde{g}_j$, $j \neq i$, are already in normal form since G was reduced to begin with. Any of the g_{ik}, though, may need to be reduced. If upon computing g_{ik} we immediately reduce it with respect to the "current" G (before computing the remaining $g_{i(k+1)}, \ldots, g_{im}$), then we must recompute the normal form of g_{ik} if one of the later $g_{ik'}$ is smaller than g_{ik}. To circumvent this situation, we order the variables so that $x_1 < x_2 < \cdots < x_m$. Thus, in Algorithm 1 G is always stored in such a way that the leading terms of all of its polynomials are in increasing order; hence, each g_{ik} need only be reduced once. In the algorithm below, $Normal(h, G)$ denotes the unique remainder of h when reduced by polynomials in G.

Lemma 2 and the above remarks imply the following theorem.

Theorem 3. *For a finite set* $V \subseteq \mathbb{F}^m$ *and a given monomial order, Algorithm 1 returns the reduced Gröbner basis for* $\mathbf{I}(V)$.

3.2. **Time complexity and comparison with current methods.** In the reduction step in line 13, we use the standard long-division technique, sometimes called Buchberger reduction. This reduction has a worst-case time complexity that may be exponential in the number m of variables. However, in practice the performance of Algorithm 1 is much better than this.

As we mentioned earlier, the methods in Buchberger and Möller (1982) [7], Marinari, Möller and Mora (1993) [21], and Abbott, Bigatti, Kreuzer and Robbiano

Algorithm 1: Vanishing ideals of distinct points

Input: $P_1, P_2, \ldots, P_n \in \mathbb{F}^m$, and a monomial order.
  Label the variables so that $x_1 < x_2 < \cdots < x_m$.
Output: G, the reduced Gröbner basis for $\mathbf{I}(P_1, \ldots, P_n)$, in increasing order.

/* *Initialization* */
$G := \{1\}$;   /* *the ith polynomial in G is denoted g_i* */

FOR k from 1 to n DO
  Find the smallest i so that $g_i(P_k) \neq 0$;
  FOR j from $i+1$ to $|G|$ DO $g_j := g_j - \frac{g_j(P_k)}{g_i(P_k)} \cdot g_i$; END FOR;
  $G := G \setminus \{g_i\}$;
  FOR j from 1 to m DO
    IF $x_j \cdot \mathrm{LT}(g_i)$ not divisible by any LT of G THEN
      Compute $h := Normal((x_j - a_j) \cdot g_i, G)$;
      Insert h (in order) into G;
    END IF;
  END FOR;
END FOR;

RETURN G.

(2000) [1] are based on Gauss elimination and have a polynomial time complexity $O(n^3 m)$. We compare our Algorithm 1 particularly with the algorithm of Marinari, Möller and Mora [21], which we designate MMM.

The Gröbner basis found via the method of O'Keeffe and Fitzpatrick [22] is minimal in the sense that the number of polynomials in the basis is the smallest possible, but the length of the polynomials computed may grow exponentially in the number m of variables. Hence, it has an exponential time complexity. For example, for 200 random points in $\mathbb{F}_5^{10}$, the largest polynomial in O'K-F's Gröbner basis typically has roughly 300 terms for *glex* order, and roughly 1500 terms for pure *lex* order. So, most of the computing time in O'K-F is taken up with dealing with large polynomials, and most of the time in Algorithm 1 involves the reduction step, *i.e.*, computing $Normal(g, G)$.

Computer experiments (for details, see [10]) indicate that as long as the number of variables is small with respect to the number of points, say $m \leq 12$ for up to 1000 points, Algorithm 1 has an advantage over both MMM and O'Keeffe-Fitzpatrick. When m is very small, say $m < 5$, the advantage of Algorithm 1 over MMM is significant: by a factor ranging from two to ten. When the number of variable climbs above 12, the linear algebra approach of MMM is faster than O'K-F and Algorithm 1. In most practical applications of coding theory, m is usually small, say $m \leq 3$.

4. Padé Approximation

The classical Padé approximation theory for univariate polynomials says that for any polynomials $f, g \in \mathbb{F}[x]$, where $\mathbb{F}$ is any field and g has degree $n > 1$, and for any positive integers n_1 and n_2 with $n_1 + n_2 = n + 1$, there are polynomials $a \in \mathbb{F}[x]$ of degree $< n_1$ and $b \in \mathbb{F}[x]$ of degree $< n_2$ so that

$$a \cdot f \equiv b \mod g, \tag{2}$$

and the ratio b/a is unique for all the solutions a and b. Furthermore, the extended Euclidean algorithm can be used to find a minimal solution a and b.

Multivariate polynomials have a similar theory. Given a function $f(x_1, \ldots, x_m)$, the generalized Padé approximation problem is to find suitable polynomials $a, b \in \mathbb{F}[x_1, \ldots, x_m]$ so that $f \equiv \frac{b}{a}$ modulo some predetermined conditions. The details of the requirements for a and b vary for different types of problems. A general approach is to consider solutions of the form

$$a \cdot f \equiv b \mod \mathbf{I}, \tag{3}$$

where $\mathbf{I} \subset \mathbb{F}[x_1, \ldots, x_m]$ is a given ideal. In the univariate case above, $\mathbf{I}$ is the ideal generated by g in $\mathbb{F}[x]$. In the multivariate case, the ideal $\mathbf{I}$ is more complicated. For different choices of ideals, (3) generalizes various forms of approximation that are studied in the literature.

The straightforward approach to finding a suitable Padé approximant is to recognize (3) as a homogeneous linear system (where the coefficients of a and b are unknowns) and to apply Gauss elimination. This linear algebra approach has cubic complexity in the number of coefficients in a and b. We measure *the degree of approximation* by the total number of coefficients in a and b. When the number of variables is small compared to the degree of approximation, a more efficient approach is via Gröbner bases [14, 20, 11]. We describe below the most recent results from [11].

We fix an arbitrary monomial order on $\mathbb{F}[x_1, \ldots, x_m]$. As mentioned in the introduction section, $\mathcal{B}(\mathbf{I})$ is a monomial basis for the quotient ring $\mathbb{F}[x_1, \ldots, x_m]/\mathbf{I}$ as a vector space over $\mathbb{F}$. For the Padé approximation problem in (3), $\mathbf{I}$ is a zero-dimensional ideal in $\mathbb{F}[x_1, \ldots, x_m]$, so the quotient ring $\mathbb{F}[x_1, \ldots, x_m]/\mathbf{I}$ is finite dimensional as a vector space over $\mathbb{F}$. We call this dimension *the degree* of $\mathbf{I}$. The degree of $\mathbf{I}$ corresponds to the degree of approximation mentioned above. The Padé approximation problem is to find certain "minimal" solutions a and b for any given monomial order.

Let z be a new variable and define

$$M_f = \{az - b \,:\, a, b \in \mathbb{F}[x_1, \ldots, x_m] \text{ satisfy } (3)\}.$$

In other words, M_f is the collection of all the polynomials in $\mathbb{F}[x_1, \cdots, x_m, z]$ that have degree at most one in z and becomes zero modulo $\mathbf{I}$ when z is replaced by f. One can check that M_f has the following properties:

(a) closed under addition, *i.e.*, if $az + b$ and $cz + d$ are in M_f then $(az + b) + (cz + d) = (a + c)z + (b + d)$ is also in M_f; and

(b) closed under multiplication by any polynomials, *i.e.*, if $az+b$ is in M_f, then $h\cdot(az+b)$ is in M_f for all $h\in\mathbb{F}[x_1,\ldots,x_m]$.

Hence, M_f forms a *module* over the ring $\mathbb{F}[x_1,\ldots,x_m]$. (A module is a kind of vector space over a ring.)

Theorem 4. *Let $\mathbf{I}$ be a zero-dimensional ideal in $\mathbb{F}[x_1,\ldots,x_m]$ of degree n. Fix a monomial order on $\mathbb{F}[x_1,\ldots,x_m]$, and denote the corresponding monomial basis by*

$$\mathcal{B}(\mathbf{I})=\{1=\mathbf{x}^{\alpha_1},\mathbf{x}^{\alpha_2},\ldots,\mathbf{x}^{\alpha_n}\},$$

ordered in increasing order. Then for any $f\in\mathbb{F}[x_1,\ldots,x_m]$ and any positive integers n_1 and n_2 with $n_1+n_2=n+1$, M_f contains a nonzero solution $az+b$ of the form

$$a=\sum_{i=1}^{n_1}a_i\mathbf{x}^{\alpha_i},\qquad b=\sum_{i=1}^{n_2}b_i\mathbf{x}^{\alpha_i}. \tag{4}$$

Furthermore, the reduced Gröbner basis for M_f under a certain term order contains a solution $az+b$ of the above form.

The main problem in the above theorem is to define an appropriate term order on polynomials of the form $az+b$ that extends any given monomial order on $\mathbb{F}[x_1,\ldots,x_m]$. The basic idea is to introduce weights on the variable z based on the weights of $\mathbf{x}^{\alpha_{n_1}}$ and $\mathbf{x}^{\alpha_{n_2}}$ imposed by the given monomial order on $\mathbb{F}[x_1,\ldots,x_m]$. For more details on this, we refer the reader to [11]. We should mention that the ratio b/a is in general not unique any more for multivariate polynomials. Also, when the monomial order used changes, the solution will often change as well.

In our coding theory application, the polynomial f is given implicitly. Given points $P_1,\ldots,P_n\in\mathbb{F}^m$ and n values $r_1,\ldots,r_n\in\mathbb{F}$, the polynomial f is defined to be any polynomial in $\mathbb{F}[x_1,\ldots,x_m]$ such that

$$f(P_i)=r_i,\quad 1\le i\le n. \tag{5}$$

That is, f is an interpolation polynomial.

In several applications, one needs to find an explicit interpolation polynomial that is smallest in a certain sense. In such a case, Algorithm 1 can be used to find a desired interpolation polynomial simply by computing the reduced Gröbner basis for the vanishing ideal of the augmented points $(P_1,r_1),\ldots,(P_n,r_n)$. The following theorem indicates how an interpolation polynomial arises in such a Gröbner basis.

Theorem 5. *Fix any monomial order on $\mathbb{F}[x_1,\ldots,x_m]$, and let G be the reduced Gröbner basis for $\mathbf{I}=\mathbf{I}(P_1,\ldots,P_n)$ and $\mathcal{B}(\mathbf{I})=\{\mathbf{x}^{\alpha_1},\ldots,\mathbf{x}^{\alpha_n}\}$, the corresponding monomial basis.*

(i) *For any $r_1,\ldots,r_n\in\mathbb{F}$, there is a unique polynomial $f=\sum_{i=1}^n f_i\mathbf{x}^{\alpha_i}$, where $f_i\in\mathbb{F}$, satisfying (5).*
(ii) *Define a new variable z and the unique monomial order on $\mathbb{F}[x_1,\ldots,x_m,z]$ that extends the given monomial order on $\mathbb{F}[x_1,\ldots,x_m]$ and that has z larger than all monomials in $x_1,\ldots,x_m$. Then the reduced Gröbner basis for $\mathbf{I}((P_1,r_1),\ldots,(P_n,r_n))$ is of the form $G\cup\{z-f\}$, where f is the unique polynomial in (i).*

In our decoding application, however, we don't need to find any interpolation polynomial f explicitly. Note that if f is an interpolation polynomial satisfying (5), then $az+b \in M_f$ if and only if $az+b$ vanishes on the points $(P_1, r_1), \ldots, (P_n, r_n)$; that is,

$$a(P_i) \cdot r_i + b(P_i) = 0, \quad 1 \leq i \leq n. \tag{6}$$

Hence, Algorithm 1 can be used to find a desired solution $az+b \in M_f$ from the augmented points $(P_1, r_1), \ldots, (P_n, r_n)$ directly without first finding an explicit form for f. One needs simply to restrict the degree of the last variable z to be at most one in the process of the algorithm. The more subtle part is to define appropriate weights on z, for details on this the reader is again referred to the paper [11]. Theorem 4 guarantees that minimal solutions $az+b \in M_f$ can indeed be computed via Gröbner basis techniques. This approach is more efficient than linear algebra approach when the number of variables is small compared to the number n of points.

5. Construction and Decoding of Linear Codes

In this section, we show how Gröbner basis techniques can be used to construct linear codes that can be decoded efficiently. These codes includes the well-known Reed-Solomon codes, Hermitian codes, and, more generally, any one-point algebraic geometry code. More detail may be found in [12].

5.1. **Construction.** For a given block length n, dimension k and alphabet size q, we use the following method to construct (n, k) linear codes over any finite field $\mathbb{F}_q$.

Choose any set of n distinct points $V = \{P_1, P_2, \ldots, P_n\}$ from $\mathbb{F}_q^m$, where m is any integer such that $q^m \geq n$. Let $\mathbf{I} = \mathbf{I}(V)$ be the vanishing ideal of V. Fix a monomial order on $\mathbb{F}[x_1, \ldots, x_m]$ (which may depend on the geometric structure of the points in V), and let the elements in the corresponding monomial basis of $\mathbf{I}$ be

$$\mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_n}$$

which are arranged in *increasing order*. Recall that these monomials are linearly independent modulo $\mathbf{I}$, which implies that the matrix

$$\begin{pmatrix} \mathbf{x}^{\alpha_1}(P_1) & \ldots & \mathbf{x}^{\alpha_1}(P_n) \\ \mathbf{x}^{\alpha_2}(P_1) & \ldots & \mathbf{x}^{\alpha_2}(P_n) \\ \vdots & & \vdots \\ \mathbf{x}^{\alpha_n}(P_1) & \ldots & \mathbf{x}^{\alpha_n}(P_n) \end{pmatrix}$$

has rank n, where $\mathbf{x}^{\alpha}(P)$ denotes the value of a monomial $\mathbf{x}^{\alpha}$ at a point P. Define $L_k \subseteq \mathbb{F}[x_1, \ldots, x_m]$ to be the linear span over $\mathbb{F}_q$ of the first k monomials, that is,

$$L_k = Span\,\{\mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_k}\} = \left\{ f(x_1, \ldots, x_m) = \sum_{i=1}^{k} f_i \mathbf{x}^{\alpha_i} : f_i \in \mathbb{F}_q \right\}. \tag{7}$$

Then we define a code $\mathcal{C}$ by

$$\mathcal{C} = \{(f(P_1), f(P_2), \ldots, f(P_n)) : f \in L_k\}.$$

Equivalently, $\mathcal{C}$ is the linear space spanned by the first k rows of the above matrix, hence $\mathcal{C}$ is an $(n, k)_q$ linear code. Hereafter, we refer to such a code as a *polynomial* *code*. The reader is reminded that this term may carry different meanings in other coding theory contexts; for example, see [18, 5, 4].

Note that there is a lot of freedom in this construction. The alphabet size q can be any prime power, the block length n can be arbitrary, likewise for the point set V and the monomial order. The error correction capability of a polynomial code depends highly on the point set and the monomial order used. As the examples below indicate, certain sets of points result in codes with large minimum distances, in fact, in maximum distance separable codes. The other extreme is also possible. Of course most randomly constructed polynomial codes have a minimum distance that lies somewhere between. In general, determining the minimum distance of these codes seems to be a difficult problem.

5.2. **Examples.** We demonstrate how several well-known classes of codes appear as special cases of our polynomial codes.

Reed-Solomon codes. Suppose $V = \mathbb{F}_q$. Then $x^q - x$ forms a Gröbner basis for$\mathbf{I}(V)$, thus $\mathcal{B} = \{1, x, \dots x^{n-1}\}$. For $k < q$, let $L_k = \{1, x, \dots x^{k-1}\}$. Then $\mathcal{C}$ is the well-known (extended) Reed-Solomon code with parameters $[n = q, k, n-k+1]_q$.

Reed-Muller codes. Suppose $V = \mathbb{F}_q^m$ where $m \geq 1$. Then $\{x_i^q - x_i : 1 \leq i \leq m\}$ is a Gröbner basis for $\mathbf{I}(V)$ under any monomial order, so

$$\mathcal{B}(\mathbf{I}) = \{x_1^{i_1} \dots x_m^{i_m} : 0 \leq i_j < q, 1 \leq j \leq m\}.$$

Let L_k be spanned by all the monomials in $\mathcal{B}(\mathbf{I})$ that have total degree $\leq r$. Then $\mathcal{C}$ is $\mathcal{R}_{\mathbb{F}_q}(r, m)$, the q-ary Reed-Muller code of order r, having parameters $[n = q^m, k, d]_q$. According to chapter five of [3],

$$k = |L_k| = \sum_{i=0}^{r} \sum_{j=0}^{m} (-1)^j \binom{m}{j} \binom{i - jq + m - 1}{i - jq},$$

and

$$d = (q - s_0)q^{m-s_1-1} - 1,$$

where $r = s_1(q-1) + s_0$, $0 \leq s_0 < q - 1$.

Hermitian codes. Suppose $q = v^2$ where v is any prime power. Consider the set V of all the points $(\alpha, \beta) \in \mathbb{F}_q^2$ that lie on the following so-called Hermitian curve

$$y^v + y = x^{v+1}.$$

One can check that for each $\alpha \in \mathbb{F}_q$ there are exactly v elements $\beta \in \mathbb{F}_q$ so that $\beta^v + \beta = \alpha^{v+1}$. Hence, V has $n = qv = v^3$ points. Note that

$$\mathbf{I}(V) = \{x^q - x, y^v + y - x^{v+1}\}.$$

We define a monomial order on $\mathbb{F}[x, y]$ using first the $(v, v+1)$-weight degree and then the degree in y (this makes y^v the leading term in $y^v + y - x^{v+1}$). Then $x^q - x$ and $y^v + y - x^{v+1}$ form a Gröbner basis for $\mathbf{I}(V)$, and the corresponding monomial basis is

$$\{x^i y^j \ : \ 0 \leq i < q, 0 \leq j < v\}. \tag{8}$$

Note that each of these monomials has different $(v, v+1)$-weighted degrees. For any $k < n$, let L_k be the linear span of the smallest k monomials in (8). Then L_k defines an (n, k) Hermitian code. The minimum distance of this code is $d = n - k + 1 - g$ for $k > 2g$ where $g = v(v-1)/2$ is the genus of the Hermitian curve.

Algebraic geometry codes. Hermitian codes are a special case of a large class of powerful codes called algebraic geometry codes. These codes are defined by points on algebraic curves (which could lie in a higher dimensional space). Their construction uses divisors and pole orders. One can show that any one-point algebraic geometry code can be converted into a polynomial code in our framework [12]. Hence, our code construction includes all the good one-point algebraic geometry codes.

5.3. **Decoding.** We describe a simple decoding procedure for polynomial codes that seems to perform well on average. Suppose $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ is a received vector. If $V = \{P_1, \ldots, P_n\}$ is the set of points in $\mathbb{F}_q^m$ that is used to define the polynomial code, then we create a set of augmented points:

$$V_r = \{(P_1, r_1), (P_2, r_2), \ldots, (P_n, r_n)\} \subset \mathbb{F}_q^{m+1}.$$

Define the vanishing module of V_r to be

$$M(V_r) = \{h = u \cdot z + v : u, v \in \mathbb{F}_q[\mathbf{x}],\ h(P_i, r_i) = 0, 1 \leq i \leq n\}.$$

Notice that if no errors occur, *i.e.*, $\mathbf{r} = \mathbf{c} = (f(P_1), \ldots, f(P_n))$, then $M(V_r) = M_f$ as defined in Section 4.

We next compute the reduced Gröbner basis G for $M(V_r)$ under the term order of Theorem 4 (described in [11]). Specifically, if $\mathbf{x}^{\alpha_k}$ is the largest monomial in L_k, then choose weights for the placeholder variable z so that z and $\mathbf{x}^{\alpha_k}$ are consecutive terms in $M(V_r)$ with $z > \mathbf{x}^{\alpha_k}$. Each polynomial in G is of the form $g = u \cdot z + v$, $u, v \in \mathbb{F}_q[x_1, \ldots, x_m]$. We are especially interested in the smallest $g_0 = u_0 \cdot z + v_0 \in G$ such that $\mathrm{LT}(g_0)$ is divisible by z.

We take this u_0 to be an error-locator polynomial. That is, if $u_0(P_i) = 0$, then we conclude that r_i is in error. More precisely, an error locator *must* evaluate to zero for each P_i such that r_i is in error, and it *may* evaluate to zero for a small number of P_i such that r_i is not in error. To understand the motivation for treating u_0 in such a way, notice that for any error-locator polynomial, say $w \in \mathbb{F}_q[x_1, \ldots, x_m]$, $w \cdot (z - f) \in M(V_r)$ since either $r_i = c_i$ implying $(z - f)(P_i, r_i) = 0$ or $r_i \neq c_i$ implying $w(P_i) = 0$. We expect the smallest such error locator to appear in one of the entries in G; in fact, we expect u_0 to be this small error-locator polynomial.

There are two scenarios in which this decoding strategy might fail to find a suitable error-locator polynomial. First, if more errors occur than the code is able to correct, then we expect that the leading term of $w \cdot (z - f)$, for any error locator w, will be too large to be in G; that is, there would be a smaller $u \cdot z + v \in M(V_r)$ with $\mathrm{LT}(u \cdot z + v)$ dividing $w \cdot z$. Secondly, if the points $P_1, \ldots, P_n$ have a particularly bad geometric structure (usually meaning a large number of the points satisfy a polynomial with small weighted degree), then we might also have the case in which every legitimate error-locator polynomial is reduced by some smaller polynomial. Incidentally, in the latter case the algorithm usually returns a nearby codeword, but not the closest codeword.

Once a suitable error locator is found, we find the Gröbner basis for $M(V_r')$, where

$$V_r' = V_r \setminus \{(P_i, r_i) : u_0(P_i) = 0\}.$$

Again, we look for the smallest polynomial with leading term divisible by z. We note that $z - f \in M(V_r')$ and, in fact, must be this small polynomial which we have found. We emphasize again that this final step depends on the existence of a suitable error-locator polynomial.

In summary the decoding process is

(1) Compute G, the reduced Gröbner basis for $M(V_r)$;
(2) Find the error-locator polynomial u_0 from G;
(3) Compute G', the reduced Gröbner basis for $M(V_r')$;
(4) Find the message polynomial f from G';

This decoding method works reasonably well in general. In certain cases it does quite well. When applied to Reed-Solomon codes, for instance, it is equivalent to the recent algorithm described in [15]. For other algebraic geometry codes, the decoder provably corrects up to $(d-1-g)/2$ errors (where g denotes the genus of the curve). In practice, the theoretical examples that cause failure in the range $((d-1-g)/2, (d-1)/2])$ rarely occur, so the decoder usually performs better than is guaranteed by the theory. Additionally, we are able to use this decoding strategy to decode a randomly constructed polynomial code (about which we know very little concerning its minimum distance). The decoding results for these random codes are more complex to describe, and we again refer the reader to [12] for details.

Finally, in practice we are sometimes interested in correcting erasures as well as errors. An erasure is simply a position in a received vector that we know has a high probability of being in error. We note that our decoding procedure is able to handle erasures without any additional time complexity—in fact, erasure information actually improves the computational performance of our decoding.

6. Final Remarks

We have presented a general framework for constructing linear codes together with a decoding algorithm. We use only the basic concepts from Gröbner basis theory which itself is more elementary than algebraic geometry. Even though we can not say much about the minimum distances of the codes constructed, our computer experiments indicate that our decoding algorithm performs reasonably welll compared with the Shannon's entropy bound. It is desirable to improve on the efficiency of our decoding algorithm. In particular, we wonder if the majority voting technique introduced by Feng and Rao [13, 9] for algebraic geometry codes can be adapted to our polynomial codes. It is not clear how this can be done, since our decoding algorithm does not compute any syndromes. Also, further work is needed to understand the minimum distances of polynomial codes and how how they depend on the geometric structure of the points that are used to define the codes. Finally, it would be of interest to see how our method can be adapted to construct quantum codes that have a natural decoding algorithm.

References

[1] J. Abbott, A. Bigatti, M. Kreuzer and L. Robbiano, Computing ideals of points, *J. Symbolic Comput.* **30** (2000), 341–356.

[2] William W. Adams and Philippe Loustaunau, *An introduction to Gröbner bases*, Graduate Studies in Mathematics, 3, American Mathematical Society, Providence, RI, 1994.

[3] E.F. Assmus, Jr. and J.D. Key, *Designs and their codes*, Cambridge Tracts in Mathematics, Vol. 103 (Second printing with corrections), Cambridge University Press, Cambridge, 1993.

[4] E.F. Assmus, Jr. and J.D. Key, Polynomial codes and finite geometries, in *Handbook of coding theory*, Vol. II (V. S. Pless, W. C. Huffman, eds.), pp. 1269–1343, North-Holland, Amsterdam, 1998.

[5] Ian F. Blake and Ronald C. Mullin, *The mathematical theory of coding*, Academic Press, New York-London, 1975.

[6] Ian Blake, Chris Heegard, Tom Høholdt and Victor Wei, Algebraic-geometry codes, *IEEE Trans. Information Theory* **44** (1998), 2596–2618.

[7] B. Buchberger and H. M. Möller, The construction of multivariate polynomials with preassigned zeros, *Computer algebra, EUROCAM '82*, pp. 24–31, Lecture Notes in Comput. Sci., vol. 144, Springer, Berlin-New York, 1982.

[8] David Cox, John Little and Donal O'Shea, *Ideals, varieties, and algorithms*, 2nd ed., Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1997.

[9] Ivan Duursma, Majority coset decoding, *IEEE Trans. Information Theory* **39** (1993), 1067–1070.

[10] Jeffrey B. Farr and Shuhong Gao, Computing Gröbner bases for vanishing ideals of finite sets of points, *preprint.*

[11] Jeffrey B. Farr and Shuhong Gao, Gröbner bases and generalized Padé approximation, *submitted.*

[12] Jeffrey B. Farr, Shuhong Gao and Daniel L. Noneaker, Construction and decoding of linear codes via Göbner bases, *in preparation.*

[13] G.-L. Feng and T. R. N. Rao, Decoding algebraic-geometric codes up to the designed minimum distance, *IEEE Trans. Information Theory* **39** (1993), 37–45.

[14] Patrick Fitzpatrick and John Flynn, A Gröbner basis technique for Padé approximation, *J. Symbolic Comput.* **13** (1992), 133–138.

[15] Shuhong Gao, A new algorithm for decoding Reed-Solomon codes, in *Communications, information and network security* (V. Bhargava, H.V. Poor, V. Tarokh and S. Yoon, eds.), Kluwer Academic Publishers, 2003, pp. 55–68.

[16] V. D. Goppa, *Geometry and Codes*, Mathematics and Its Applications, Vol. 24, Soviet Series, Kluwer Academic Publishers, Dordrecht, 1988.

[17] Tom Høholdt, Jacobus H. van Lint and Ruud Pellikaan, Algebraic geometry codes, in *Handbook of Coding Theory*, Vol. I (V. S. Pless, W. C. Huffman, eds.), pp. 871–961, North Holland, Amsterdam, 1998.

[18] Tadao Kasami, Shu Lin and W. Wesley Peterson, Polynomial codes, *IEEE Trans. Information Theory* **14** (1968), 807–814.

[19] M. Kreuzer and L. Robbiano, *Computational Commutative Algebra 1*, Springer-Verlag, Berlin, 2000.

[20] John B. Little, David Ortiz, Ricardo Ortiz-Rosado, Rebecca Pablo and Karen Ríos-Soto, Some remarks on Fitzpatrick and Flynn's Göbner basis technique for Padé approximation, *J. Symbolic Comput.* **35** (2003), 451–461.

[21] M. G. Marinari, H.M. Möller and T. Mora, Gröbner bases of ideals defined by functionals with an application to ideals of projective points, *Appl. Algebra Engrg. Comm. Comput.* **4** (1993), no. 2, 103–145.

[22] Henry O'Keeffe and Patrick Fitzpatrick, Gröbner basis solutions of constrained interpolation problems, Fourth special issue on linear systems and control, *Linear Algebra Appl.* **351/352** (2002), 533–551.

[23] Reinhard Laubenbacher and Brandilyn Stigler, A computational algebra approach to the reverse engineering of gene regulatory networks, *preprint*, 2003.

[24] G. Pistone, E. Riccomagno, H. P. Wynn, *Algebraic Statistics: Computational Commutative Algebra in Statistics*, Monographs on Statistics & Applied Probability 89, Chapman & Hall/CRC, 2001.

[25] Henning Stichtenoth, *Algebraic Function Fields and Codes*, Springer-Verlag, Berlin, 1993.

Department of Mathematical Sciences, Clemson University, Clemson, SC 29634-0975, USA *E-mail address*: {jeffref, sgao}@ces.clemson.edu