On Representations of Algebraic-Geometric Codes

Venkatesan Guruswami Madhu Sudan*

MIT Laboratory for Computer Science
200 Technology Square
Cambridge, MA 02139.
Email: venkat@theory.lcs.mit.edu, madhu@mit.edu.

Abstract

We show that all algebraic-geometric codes possess a succinct representation that allows for the list decoding algorithms of [9, 6] to run in polynomial time. We do this by presenting a root-finding algorithm for univariate polynomials over function fields when their coefficients lie in finite-dimensional linear spaces, and proving that there is a polynomial size representation given which the root finding algorithm runs in polynomial time.

1 Introduction

Algebraic function fields, and arithmetic over these fields, play a central role in coding theory. The famed algebraic-geometry codes (AG codes) are based on the structure of these fields and lead to the best known family of error-correcting codes, over large enough alphabets. Further, decoding algorithms for these codes also rely strongly on the algebraic properties of these function fields. In particular, the rely on the ability to perform certain operations over these fields. These operations include basic field operations such as addition and multiplication, but also some non-trivial operations such as evaluating functions at "places", and finding roots of polynomials over these fields.

In view of the central role played by the operation of root-finding in the task of decoding AG codes, several recent works have examined the complexity of this task for classes of function fields. These works include those of Gao and Shokrollahi [4], Høholdt and Nielsen [8], Matsumoto [7], Wu and Siegel [13], and Augot and Pecquet [1]. The techniques developed are quite general, however the explicitly stated results are not and fall into one of the following categories: (a) Either these algorithms work only for specific function fields; for example the algorithms in [8] work for function fields of Hermitian curves, and those in [4] work for function fields of nonsingular plane algebraic curves, or (b) as in [1, 13], the algorithms reduce the decoding task to certain "basic" algorithmic tasks on function fields, and it is not clear how to perform these basic tasks efficiently for *every* function field. The sole exception may be the work of Matsumoto [7], who, independent of our work, implicitly suggests a completely general solution to the problem of root-finding over all fields.

Closer examination of these works reveals an even more important question: What exactly is known about the underlying function field? How are the elements of the field represented and how are elementary operations performed? This leads to the essential bottleneck preventing a unified

*Supported in part by an MIT-NEC Research Initiation Award, a Sloan Foundation Fellowship and NSF Career Award CCR-9875511 and NSF Award CCR-9912342.

presentation of results: One needs a generic method to represent the elements of an algebraic function field so that (1) these representations are short for the elements of interest in the construction of the algebraic-geometric codes; and (2) basic operations are efficient under this representation.

For example, a typical example of an algebraic function field is the field of fractions obtained from the ring of functions in $\mathbb{F}_q[X,Y]/(f(X,Y))$, where $f(X,Y)$ is an absolutely irreducible curve. It is usual to represent elements of this domain by rational functions in X and Y. In such a representation, the size of an element is polynomially related to its degree. For applications to algebraic-geometric codes, this representation is sufficient to establish property (1) above. However, it is not immediate that this representation can lead to efficient algorithms for evaluations, or root-finding, and thus Property (2) above may not be satisfied. For this part one needs to be more explicit about the polynomial f. Furthermore, as one considers more general classes of function fields, it is not clear that this property can always be satisfied by this representation. As a consequence some of the recent works on (list) decoding of algebraic-geometric codes (henceforth referred to as AG-codes) either remain specific to certain codes [4, 8], or are just described as reductions to other algebraic problems (as in the case of [6]).

Here we resolve this issue, by reformulating the problem. We show how it is possible to adopt an alternate description for elements of the function field which remains terse, while allowing basic tasks to be carried out efficiently. We focus on the most complex task arising from list decoding - namely that of finding roots of polynomials defined over function fields. For this task, we show how the algorithms from Gao and Shokrollahi [4] or the similar algorithm of Høholdt and Nielsen [8]) can be implemented so as to work in our representation.

Since all other operations needed to implement the recent list decoding algorithms are also easy to carry out in our representation, we also are able to present a compact and general theorem about list decoding of AG-codes.

2 Algebraic-geometric codes: Preliminaries

We now abstract the main notions associated with the theory of algebraic function fields that will be important to us. The interested reader may find further details in [10, 5]. In what follows we assume familiarity with the basic notions of field extensions.

Places, Valuations, and Degrees: A function field $K/\mathbb{F}_q$ has a set of *places* $\mathbb{P}_K$ and the associated set of *valuations*, given by a valuation map $v : \mathbb{P}_K \times K \to \mathbb{Z} \cup \{\infty\}$. The exact definition of these notions can be found, for instance, in [10]; we only abstract some properties relevant to us below. As is normal practice, for each $P \in \mathbb{P}_K$, we denote by $v_P : K \to \mathbb{Z} \cup \{\infty\}$, the map $v_P(\cdot) = v(P, \cdot)$ which tells how many zeroes or poles a given function has at P (with the convention $v_P(0) = \infty$). The valuation v_P at any place satisfies the following properties:

(a) $v_P(a) = \infty$ iff $a = 0$ and $v_P(a) = 0$ for all $a \in \mathbb{F}_q \setminus \{0\}$.

(b) $v_P(ab) = v_P(a) + v_P(b)$ for all $a, b \in K \setminus \{0\}$.

(c) $v_P(a + b) \geq \min\{v_P(a), v_P(b)\}$.

Associated with every place is a *degree* abstracted via the map $\deg : \mathbb{P}_K \to \mathbb{Z}^+$. The degree, $\deg(P)$, of any place P is a positive integer and intuitively means the following: when we pick a function $f \in K$ which has no poles at P and "evaluate" it at P, we get a value in the field $\mathbb{F}_{q^{\deg(P)}}$. Thus places of degree one correspond to *rational points* on the curve.

Evaluations of functions at places: We can abstract the notion of *evaluation* of elements of the function field at the places by a map $\mathsf{eval}: K \times \mathbb{P}_K \to \bar{\mathbb{F}}_q \cup \{\infty\}$ (here $\bar{\mathbb{F}}_q$ is the algebraic closure of $\mathbb{F}_q$). This map has the following properties:

(i) $\mathsf{eval}(f, P) = \infty$ iff $v_P(f) < 0$, and $\mathsf{eval}(f, P) = 0$ iff $v_P(f) > 0$ for every $P \in \mathbb{P}_K$ and $f \in K$.

(ii) If $f \in K$, $P \in \mathbb{P}_K$ and $v_P(f) \geq 0$, then $\mathsf{eval}(f, P) \in \mathbb{F}_{q^{\deg(P)}}$.

(iii) The map eval respects field operations; i.e., if $v_P(f_1) \geq 0$ and $v_P(f_2) \geq 0$, then $\mathsf{eval}(f_1 + f_2, P) = \mathsf{eval}(f_1, P) + \mathsf{eval}(f_2, P)$, and $\mathsf{eval}(f_1 * f_2, P) = \mathsf{eval}(f_1, P) * \mathsf{eval}(f_2, P)$.

Divisors: The *divisor group* $\mathcal{D}_K$ of the function field K is a free abelian group on $\mathbb{P}_K$. An element D of $\mathcal{D}_K$ is thus represented by the formal sum $\sum_{P \in \mathbb{P}_K} a_P P$ where $a_P = 0$ for all but finitely many P; we say $D \succeq 0$ if $a_P \geq 0$ for all $P \in \mathbb{P}_K$. The support of a divisor D, denoted $\mathrm{supp}(D)$, is the (finite) set $\{P \in \mathbb{P}_K : a_P \neq 0\}$. The degree map extends naturally to a homomorphism $\deg : \mathcal{D}_K \to \mathbb{Z}$ as $\deg(\sum_P a_P P) = \sum_P a_P \deg(P)$.

For every $f \in K \setminus \{0\}$, there is an associated divisor, called the principal divisor and denoted (f), which is defined by $(f) = \sum_P v_P(f) P$. The following result is well known, and just states that every non-zero function in the function field has an equal number of zeroes and poles.

Theorem 1 *For any function field $K/\mathbb{F}_q$ and any $f \in K \setminus \{0\}$, $\deg((f)) = 0$.*

For every divisor $D \in \mathcal{D}_K$, one can define the linear space of functions $\mathcal{L}(D)$ as $\mathcal{L}(D) = \{f \in K : (f) + D \succeq 0\}$. It is known that for any divisor $D \succeq 0$, $\mathcal{L}(D)$ is a finite-dimensional vector space over $\mathbb{F}_q$ and $\dim(\mathcal{L}(D)) \leq 1 + \deg(D)$ (see [10] for a proof). A lower bound on $\dim(\mathcal{L}(D))$ is given by the celebrated Riemann-Roch theorem for function fields, which states that there is a non-negative integer g (called "genus"), that depends only on $K/\mathbb{F}_q$, such that $\dim(\mathcal{L}(D)) \geq \deg(D) - g + 1$.

Algebraic-geometric codes: We are now ready to define the notion of an AG-code. Let $K/\mathbb{F}_q$ be an algebraic function field of genus g, let $Q, P_1, P_2, \ldots, P_n$ be *distinct* places of degree one in $\mathbb{P}_K$, and let $G = P_1 + P_2 + \cdots + P_n$ and $D = \alpha Q$ be divisors of $K/\mathbb{F}_q$ (note that $\mathrm{supp}(G) \cap \mathrm{supp}(D) = \emptyset$).

The algebraic-geometric code $\mathcal{C}_{\mathcal{L}}(G, D) = \mathcal{C}_{\mathcal{L}}(G, \alpha, Q)$ is defined by[1]

$$\mathcal{C}_{\mathcal{L}}(G, \alpha, Q) := \{(\mathsf{eval}(f, P_1), \ldots, \mathsf{eval}(f, P_n)) : f \in \mathcal{L}(\alpha Q)\} \subseteq \mathbb{F}_q^n \ .$$

(Note that $\mathsf{eval}(f, P_i) \in \mathbb{F}_q$ since $v_{P_i}(f) \geq 0$ and $\deg(P_i) = 1$.) The following Proposition follows from Theorem 1 and the Riemann-Roch theorem, and quantifies the parameters of these codes.

Proposition 2 *Suppose that $\alpha < n$. Then $\mathcal{C}_{\mathcal{L}}(G, \alpha, Q)$ is an $[n, k, d]_q$ code with $k = \dim(\mathcal{L}(\alpha Q)) \geq \alpha - g + 1$ and $d \geq n - \alpha$ (thus $k + d \geq n + 1 - g$).*[2]

3 Representation Issues

As mentioned earlier, algorithms that involve function fields, and in particular decoding algorithms for AG-codes raise several issues on how to represent elements from the function field K and the places $\mathbb{P}_K$. Specifically, we would like to perform the following operations efficiently: (i) Given two elements $x, y \in K$, compute their sum and product in K; (ii) Given $f \in K$ and $P \in \mathbb{P}_K$, compute

[1] It is clear that the defined space is a linear space.

[2] The notation $[n, k, d]_q$ code stands, as usual, for a code over $\mathbb{F}_q$ of blocklength n, rate k and minimum distance d.

the zero or pole order of f at P, and compute $\mathsf{eval}(f, P)$; and (iii) Given a divisor $D \succeq 0$, compute a basis for the vector space $\mathcal{L}(D)$ over $\mathbb{F}_q$.

First of all, the function field K is an infinite set, so one cannot assume that operations in K (like sum and product) are unit operations. Instead one must fix a representation for the elements and one must give explicit algorithms to perform these operations that are efficient with respect to the size of the representation of an element. A natural representation to consider is to express elements of K as ratio of two homogeneous multivariate polynomials. For this representations, the field operations in K can be done in time polynomial in the sum of degrees of the respective polynomials. However, for question (iii) above, it is *not* known whether for general function fields there always exists a basis for $\mathcal{L}(D)$ over $\mathbb{F}_q$ with a succinct representation (i.e. one of size polynomial in $\deg(D)$) as the ratio of polynomials (for example, see [9] for a discussion).

For applications to decoding, one does not need to work with all of K, and instead focuses attention only on elements in $\mathcal{L}(D)$ for some divisor $D \succeq 0$. This allows us the option of representing elements of $\mathcal{L}(D)$ as vectors in $\mathbb{F}_q^{\dim(\mathcal{L}(D))}$ which represent their coordinates with respect to some *fixed* basis of $\mathcal{L}(D)$ over $\mathbb{F}_q$. Since $\dim(\mathcal{L}(D)) \leq \deg(D) + 1$, this representation will be small provided $\deg(D)$ is small. Indeed, the first interpolation step in the list decoding algorithm of [6] was shown to be efficiently implementable under this representation. This will also be the representation we use here, though in order to perform the root-finding step, we augment this representation suitably. Before explaining this, we formally describe the basic root-finding task that we wish to solve, and describe an algebraic algorithm along the lines of [8, 4] to solve it. We then discuss the representation issues this algorithm motivates.

Procedure ROOT-FIND$_{(K,D)}(H)$

Input: A degree m polynomial $H = \sum_{i=0}^{m} a_i T^i \in K[T]$ where each $a_i \in \mathcal{L}(D)$.

Output: All roots of H that lie in $\mathcal{L}(D)$.

1. "Reduce" H modulo a place $R \in \mathbb{P}_K$ of large enough degree, say r (i.e., compute $b_i = \mathsf{eval}(a_i, R)$ for $0 \leq i \leq m$ and consider the polynomial $P = \sum_{i=0}^{m} b_i Y^i \in \mathbb{F}_{q^r}[Y]$).

2. Compute the roots, say $\alpha_1, \ldots, \alpha_t$, of P that lie in $\mathbb{F}_{q^r}$ using a root-finding algorithm for finite fields.

3. For each α_j, $1 \leq j \leq t$, "find" $\beta_i \in \mathcal{L}(D)$ if any such that $\mathsf{eval}(\beta_i, R) = \alpha_i$.

The tricky issue in the above algorithm is that we need to "evaluate" an $f \in \mathcal{L}(D)$ at some place $R \in \mathbb{P}_K$ of interest. To aid this we "represent" a place R by the values $\mathsf{eval}(\phi_i, R)$, for $1 \leq i \leq p$ where $p = \dim(\mathcal{L}(D))$ and $\phi_1, \ldots, \phi_p$ is a basis of $\mathcal{L}(D)$ over $\mathbb{F}_q$. Together with the representation of any element of $\mathcal{L}(D)$ as a linear combination of the ϕ_i's this clearly enables us to evaluate any element of $\mathcal{L}(D)$ at R. Since each $\mathsf{eval}(\phi_i, R) \in \mathbb{F}_{q^r}$ where $r = \deg(R)$, one can "represent" R, for purposes of evaluation by members of $\mathcal{L}(D)$, as an element of $\mathbb{F}_{q^r}^p$. (We assume that some standard representation of elements of $\mathbb{F}_{q^r}$.)

It should be somewhat clear that given these representations, the algebraic procedures discussed at the beginning of this section can in fact be turned into efficient algorithms. The next section proves formally that this is indeed the case.

4 Efficient Root-finding

In this section we describe the root-finding algorithm from the previous section formally and carefully taking all representation issues into account. We also prove the correctness of the algorithm.

The fact that it runs in polynomial time given the representation it assumes will be clear from the description.

Considering the application to list decoding in mind, we only need to solve special instances of the univariate root-finding problem. Namely, we assume the input polynomial $H \in K[T]$ of degree m has all its coefficients in a linear subspace $\mathcal{L}(D)$ for some divisor $D \succeq 0$ of K, and we only seek roots that lie in $\mathcal{L}(D)$. (For applications to AG-codes, the divisor D is actually a *one-point* divisor, i.e., is of the form ℓQ for some place Q of degree one, and moreover we will only be interested in roots that lie in $\mathcal{L}(\alpha Q)$ for some $\alpha \le \ell$. We, however, present a root-finding algorithm that works for any divisor, as this could be of independent interest.) In addition to this "uniform" input H, the algorithm also uses a non-uniform input, namely a place R of large degree, which only depends on D and *does not* depend on the degree of the input polynomial.

Instead of finding only roots that lie in $\mathcal{L}(D)$, we will more generally find all roots $\alpha \in K$ such that α has no poles outside $\mathrm{supp}(D)$. We first prove a simple lemma that limits the number of poles any such root can have at a place in $\mathrm{supp}(D)$.

Lemma 3 *Let $H \in K[T]$ be a non-zero polynomial all of whose coefficients lie in $\mathcal{L}(D)$ for some divisor $D \succeq 0$, and let $\alpha \in K$ be a root of H such that α has no poles outside* $\mathrm{supp}(D)$. *Then in fact $\alpha \in \mathcal{L}(D')$ where $D' = \sum_{P \in \mathrm{supp}(D)} \frac{\deg(D)}{\deg(P)} P$. (Note that* $\deg(D') = \deg(D)|\mathrm{supp}(D)|$*.)*

Proof: Let α be a root of H. We prove that for any $P \in \mathrm{supp}(D)$, $v_P(\alpha) \ge -\frac{\deg(D)}{\deg(P)}$, and this will clearly imply the claimed result. Let $H[T] = a_m T^m + \ldots + a_1 T + a_0$ where each $a_j \in \mathcal{L}(D)$, and let $D = \sum_{R \in \mathrm{supp}(D)} e_R R$ where each $e_R > 0$. Clearly $v_P(a_j \alpha^j) \ge -e_P + j v_P(\alpha)$ since $v_P(a_j) \ge -e_P$. If $v_P(\alpha) \ge 0$ we are done, so assume $v_P(\alpha) < 0$. Hence

$$v_P(H(\alpha) - a_m \alpha^m) = v_P\Big(\sum_{j=0}^{m-1} a_j \alpha^j\Big) \ge -e_P + (m-1) v_P(\alpha) \ . \tag{1}$$

We now upper bound $v_P(a_m)$. Since $\deg((a_m)) = 0$ and $a_m \in \mathcal{L}(D)$, we have

$$\sum_{R \in \mathrm{supp}(D)} v_R(a_m) \deg(R) \le 0,$$

and this together with $v_R(a_m) \ge -e_R$ for every R gives

$$v_P(a_m) \deg(P) \le \sum_{R \in \mathrm{supp}(D) \setminus P} e_R \deg(R) = \deg(D) - e_P \deg(P) \ . \tag{2}$$

Thus we have

$$v_P(a_m \alpha^m) \le \frac{\deg(D)}{\deg(P)} - e_P + m v_P(\alpha) \ . \tag{3}$$

Since $H(\alpha) = 0$, we must have $v_P(a_m \alpha^m) = v_P(H(\alpha) - a_m \alpha^m)$. Using Equations (1) and (3) this gives $-e_P + (m-1) v_P(\alpha) \le \frac{\deg(D)}{\deg(P)} - e_P + m v_P(\alpha)$ which gives $v_P(\alpha) \ge -\frac{\deg(D)}{\deg(P)}$, as desired. $\Box$

We next state two other simple lemmas which are necessary for the correctness of the algorithm.

Lemma 4 *For any function field K, there exists a place of degree m in $\mathbb{P}_K$ for every large enough integer m.*

Proof: By the Hasse-Weil inequality, the number N_m of places of degree m in $\mathbb{P}_K$ satisfies $|N_m - q^m - 1| \leq 2gq^{m/2}$ where g is the genus of the function field K. Hence if $m \geq m_0$ where m_0 is the smallest integer that satisfies $\frac{q^{m_0}-1}{2q^{m_0/2}} > g$, then $N_m \geq 1$. $\square$

Lemma 5 *If $f_1, f_2 \in \mathcal{L}(A)$ for some divisor $A \succeq 0$ and $\mathsf{eval}(f_1, R) = \mathsf{eval}(f_2, R)$ for some place R with $\deg(R) > \deg(A)$, then $f_1 = f_2$.*

Proof: Suppose not, so that $f_1 - f_2 \neq 0$. Then, by Theorem 1, $\deg((f_1 - f_2)) = 0$. But $f_1 - f_2 \in \mathcal{L}(A)$ and $v_R(f_1 - f_2) \geq 1$, so that $\deg((f_1 - f_2)) \geq \deg(R) - \deg(A) > 0$, a contradiction. Hence $f_1 = f_2$. $\square$

We are now ready to formally present our root-finding algorithm.

Algorithm ROOT-FIND$_{(K,D)}(H)$

Non-uniform input: (This depends only on D and is independent of the actual input.) A place $R \in \mathbb{P}_K$ such that $\deg(R) = r > \deg(D)|\mathrm{supp}(D)|$ (such a place necessarily exists; see Lemma 4). The place R is represented as an s-tuple $(\zeta_1^R, \ldots, \zeta_s^R)$ over $\mathbb{F}_{q^r}$ comprising of evaluations of the basis functions $\mathcal{B}' = \{\phi_1, \phi_2, \ldots, \phi_p, \phi_{p+1}, \ldots, \phi_s\}$ of $\mathcal{L}(D')$ at R. Here $D' = \sum_{P \in \mathrm{supp}(D)} \frac{\deg(D)}{\deg(P)} P$, $s = \dim(\mathcal{L}(D'))$, and the basis $\mathcal{B}'$ *extends* a basis $\mathcal{B} = \{\phi_1, \ldots, \phi_p\}$ of $\mathcal{L}(D)$.

Input: A polynomial $H = a_0 + a_1 T + \cdots + a_m T^m$ of degree m in $K[T]$ where each $a_i \in \mathcal{L}(D)$ for some divisor D of K. Each a_i is presented as a p-tuple $(a_{i1}, \ldots, a_{ip})$ over $\mathbb{F}_q$ where $p = \dim(\mathcal{L}(D))$, $a_{ij} \in \mathbb{F}_q$ (this is to interpreted as $a_i = \sum_{j=1}^{p} a_{ij}\phi_j$ where $\mathcal{B} = \{\phi_1, \phi_2, \ldots, \phi_p\}$ is a basis of $\mathcal{L}(D)$ over $\mathbb{F}_q$).

Output: All the roots of H that lie in K and have no poles outside the at places in $\mathrm{supp}(D)$. (By Lemma 3, all such roots lie in $\mathcal{L}(D')$.)

Step 1: Reduce H modulo R to obtain a polynomial $h \in \mathbb{F}_{q^r}[T]$: i.e., for $0 \leq i \leq m$, compute

$$b_i = \mathsf{eval}(a_i, R) = \sum_{j=1}^{p} a_{ij} \zeta_j^R \in \mathbb{F}_{q^r},$$

and set $h[T] = b_0 + b_1 T + \cdots + b_m T^m$.

Step 2: Find all the (distinct) roots $\alpha_1, \alpha_2, \ldots, \alpha_t$ of h that lie in $\mathbb{F}_{q^r}$ using a standard root finding algorithm for finite fields. (This can be accomplished in deterministic $\mathrm{poly}(q, r)$ time by an algorithm due to Berlekamp [2].)

Step 3 (Recovering the original roots): For each $\alpha_i \in \mathbb{F}_{q^r}$ such that $h(\alpha_i) = 0$ "find" the *unique* $\beta_i \in \mathcal{L}(D')$ such that $\mathsf{eval}(\beta_i, R) = \alpha_i$, in terms of its coefficients $c_{ij} \in \mathbb{F}_q$ with respect to the basis $\mathcal{B}'$ of $\mathcal{L}(D')$. (Recall that $D' = \sum_{P \in \mathrm{supp}(D)} \frac{\deg(D)}{\deg(P)} P$ and thus $\deg(D') = \deg(D)|\mathrm{supp}(D)|$, so Lemma 5 proves that such a β_i, if any, is unique.) For each i, the c_{ij}'s can be found by solving $\sum_{j=1}^{s} c_{ij}\zeta_j^R = \alpha_i$ which can be viewed as a linear system of equations over $\mathbb{F}_q$ (by fixing some representation of elements of $\mathbb{F}_{q^r}$ over $\mathbb{F}_q$).

Step 4: Output the list of roots $\{\beta_1, \ldots, \beta_t\}$.

It is clear, given Lemmas 3, 4 and 5, that the algorithm correctly finds all roots of H that have no poles outside $\mathrm{supp}(D)$. Moreover, it clearly runs in polynomial time given the non-uniform input. We thus get:

Theorem 6 *There is an efficient root-finding algorithm that, for any function field K and any divisor $D \succeq 0$, given an "advice" that depends only on D and is of size polynomial in* $\deg(D)$*, finds, in* $\mathrm{poly}(m, \deg(D))$ *time, the roots of any degree m polynomial in $K[T]$ whose coefficients all lie in $\mathcal{L}(D)$.*

Since root-finding is the main algorithmic step in the list decoding algorithm of [6] and the other steps in the decoding algorithm (like interpolation) can be performed efficiently under our representation, we conclude the main result from our work:

Theorem 7 (Main theorem) *For* every *algebraic-geometric code $\mathcal{C}_{\mathcal{L}}(G, \alpha, Q)$ of block length n and designed distance d, there is a representation of the code of size polynomial in n, such that given this representation list decoding from up to $n - \sqrt{n(n-d)}$ errors can be done in polynomial time.*

5 Conclusions

We have shown that AG-codes admit a representation given which the list decoding algorithm of [6] runs in polynomial time. It would be interesting to examine whether, for specific AG-codes which beat the Gilbert-Varshamov bound, say the codes based on the Garcia-Stichtenoth tower of function fields [5], this representation can also be *found* in polynomial time.

Acknowledgments

We thank Tom Høholdt and the anonymous referees for useful suggestions and pointers to related work.

References

[1] D. Augot and L. Pecquet. A Hensel lifting to replace factorization in list decoding of algebraic-geometric and Reed-Solomon codes. *IEEE Trans. Info. Theory*, Vol. 46, Nov. 2000, pp. 2605-2613.

[2] E. R. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computations*, 24 (1970), pp. 713-735.

[3] P. Elias. List decoding for noisy channels. *Wescon Convention Record*, Part 2, Institute of Radio Engineers (now IEEE), pp. 94-104, 1957.

[4] S. Gao and M. A. Shokrollahi. Computing roots of polynomials over function fields of curves. In *Proceedings of the Annapolis Conference on Number Theory, Coding Theory, and Cryptography*, 1999.

[5] A. Garcia and H. Stichtenoth. Algebraic function fields over finite fields with many rational places. *IEEE Trans. on Info. Theory*, 41 (1995), pp. 1548-1563.

[6] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and Algebraic-geometric codes. *IEEE Trans. on Information Theory*, 45 (1999), pp. 1757-1767. Preliminary version appeared in *Proc. of FOCS'98*.

[7] R. Matsumoto. *On the second step in the Guruswami-Sudan list decoding algorithm for AG-codes.* Technical Report of IEICE, pp. 65-70, 1999.

[8] R. R. Nielsen and T. Høholdt. Decoding Hermitian codes with Sudan's algorithm. In *Proceedings of AAECC-13, LNCS 1719, Springer-Verlag*, 1999, pp. 260-270.

[9] M. A. Shokrollahi and H. Wasserman. List decoding of algebraic-geometric codes. *IEEE Trans. on Information Theory*, Vol. 45, No. 2, March 1999, pp. 432-437.

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

[11] M. Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180-193, March 1997.

[12] J. M. Wozencraft. List Decoding. *Quarterly Progress Report*, Research Laboratory of Electronics, MIT, Vol. 48 (1958), pp. 90-95.

[13] Xin-Wen Wu and P. H. Siegel. Efficient list decoding of algebraic geometric codes beyond the error correction bound. In *Proc. of International Symposium on Information Theory*, June 2000.