DECODING OF DIFFERENTIAL AG CODES

KWANKYU LEE
Department of Mathematics Education
Chosun University
Gwangju 61452, Korea

(Communicated by Iwan Duursma)

Abstract. The interpolation-based decoding that was developed for general evaluation AG codes is shown to be equally applicable to general differential AG codes. A performance analysis of the decoding algorithm, which is parallel to that of its companion algorithm, is reported. In particular, the decoding capacities of evaluation AG codes and differential AG codes are seen to be interrelated symmetrically. As an interesting special case, a decoding algorithm for classical Goppa codes is presented.

1. Introduction

Let $X$ be a smooth geometrically irreducible projective curve defined over a finite field $\mathbb{F}$ of genus $g$. Let $\mathbb{F}(X)$ and $\Omega_X$ denote the function field and the module of differentials of $X$ respectively. Let $P_1, P_2, \ldots, P_n$ be distinct rational points on $X$, and $D = P_1 + P_2 + \cdots + P_n$. Let $G$ be an arbitrary divisor on $X$, whose support is disjoint from that of $D$. Recall that $L(G) = \{ f \in \mathbb{F}(X) : (f) + G \geq 0 \}$ and $\Omega(G) = \{ \omega \in \Omega_X : (\omega) \geq G \}$. Then Goppa’s famous two codes [7] are defined by

$$C_L(D,G) = \{ (f(P_1), f(P_2), \ldots, f(P_n)) : f \in L(G) \}$$

and

$$C_\Omega(D,G) = \{ (\text{res}_{P_1}(\omega), \text{res}_{P_2}(\omega), \ldots, \text{res}_{P_n}(\omega)) : \omega \in \Omega(-D + G) \},$$

which are respectively called evaluation AG code and differential AG code. It is well-known that $C_\Omega(D,G) = C_L(D,G)^\perp$, whose proof (see [12]) requires the Riemann-Roch theorem.

Theorem 1.1 (Riemann-Roch). Let $A$ be a divisor on $X$. Then

$$\dim L(A) = \deg(A) + 1 - g + \dim \Omega(A).$$

Moreover if $\deg(A) \geq 2g - 1$, then $\dim \Omega(A) = 0$, and if $\deg(A) < 0$, then $\dim L(A) = 0$.

Though the two kinds of AG codes were equally created, their historical development has been somewhat unbalanced. The target of intensive researches on the bounds on the minimum distance and decoding algorithms up to the bounds were usually the differential code or rather the dual code to the evaluation code. Thus

2010 Mathematics Subject Classification: Primary: 94B35, 94B27; Secondary: 94B65.
Key words and phrases: Differential AG code, decoding algorithm, interpolation, Gröbner basis, numerical semigroup.

The author was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2009714).
all known decoding algorithms for differential AG codes decode $C_L(D,G)\perp$ rather than $C_\Omega(D,G)$ itself in the sense that they work with the data called syndromes defined using functions in $L(G)$. See [3, 2, 11, 8, 1] and many references therein. The bias is also reflected on the terms like primal AG code and dual AG code, which mean evaluation AG code and differential AG code respectively. In this respect, the recent result [10] on the unique decoding algorithm for evaluation AG codes is against to the trend and implies that the duality is not essential for decoding and for bounding the minimum distance of $C_L(D,G)$. This makes one conceive of a decoding algorithm for $C_\Omega(D,G)$ that likewise does not rely on the duality, syndromes, or the space $L(G)$.

In this paper, we present a fast unique decoding algorithm for general differential AG codes, which does not rely on the duality and does not use syndromes defined by the functions in $L(G)$. Essentially it is the interpolation-based decoding algorithm for $C_L(D,G)$ of [10] rewritten for the differential AG code $C_\Omega(D,G)$ based on the same principle ideas but using data derived from the relevant space of differentials. Specifically, the ring $R$ defined in this paper (1) is the same $R$ defined in [10], but instead of $R$-module $\bar{R} \subset \mathbb{F}(X)$, here we define another $R$-module $\mathcal{W} \subset \Omega_X$. Thus the decoding algorithm for differential AG codes works with the polynomials in $R \oplus \mathcal{W}$ while the algorithm for evaluation AG codes works in $R \oplus \bar{R}$. These changes aside, the basic principles underlying both decoding algorithms are exactly the same. Thus we achieve an equal and symmetric treatment of Goppa’s two codes in decoding and bounding the minimum distance.

In Section 2, we set up the algebraic framework in which the decoding algorithm works. In Section 3, we present the decoding algorithm after a brief explanation of its structure. In Section 4, we report a performance analysis. It will be clearly seen that the framework and the algorithm itself resemble the corresponding ones in [10] so closely that, to avoid repetition, we do not prove that the algorithm works correctly nor provide the proofs for the assertions about performance analysis, but instead refer the reader to [10] for almost verbatim details of missing proofs. In Section 4.3, we will see that decoding capacities of both decoding algorithms are interrelated. In Section 5, we give an explicit example. In Section 6, we consider decoding of Goppa codes. Recall that Goppa codes are subfield subcodes of differential AG codes on the projective line. Hence as a special case, we obtain a decoding algorithm for classical Goppa codes. This algorithm is interesting because Goppa codes are the main workhorse in the McEliece code-based cryptosystem and the speed of decryption is largely dependent on the efficiency of its decoding algorithm.

2. Preliminaries

We assume the existence of a rational point $Q$ distinct from the points in the support of $D$. Let

$$R = \bigcup_{s=0}^{\infty} L(sQ) \subset \mathbb{F}(X).$$

For $f \in R$, let $\rho(f) = -v_Q(f)$. The Weierstrass semigroup at $Q$ is then

$$\Lambda = \{\rho(f) : f \in R\} = \{\lambda_0, \lambda_1, \lambda_2, \ldots\} \subset \mathbb{Z}_{\geq 0},$$

which is a numerical semigroup whose number of gaps, the positive integers not in $\Lambda$, is the genus $g$ of $X$. Let $\gamma$ be the smallest positive nongap, and let $\rho(x) = \gamma$ for some $x \in R$. For each $0 \leq i < \gamma$, let $a_i$ be the smallest nongap such that
integers, for each $0 \leq b < i < \gamma$.
The integers in $\bar{\Omega}$ will also be called nongaps. As $\bar{\Omega}$ contains all large enough $F$-solutions, the weighted degree of a polynomial $f_{\lambda}$, the leading monomial, and the leading coefficient, with respect to $Rz$, is defined by $\delta(f_{\lambda}) = \sum_i x_i.$

So we can regard the elements of $\rho(\varphi_{\lambda})$ as a free module of rank 2. Let $\bar{\Omega} = \delta(\bar{\omega})$. Notice that the ring $R$ and the numerical semigroup $\Lambda$ are the same as defined in [10]. Now come the definitions which are new but correspond to $\bar{\lambda}$. In particular, we have

$\delta_s(f_{\lambda} + \omega) = \max\{\rho(f_{\lambda} + s), \delta(\omega)\}$. In particular, we have $\delta_s(f_{\lambda} + \omega) = \gamma k + a_i + s$, $\delta_s(x^k \bar{\omega}_i) = \delta(x^k \bar{\omega}_i) = \gamma k + b_i$. Then $\delta_s$ induces the weighted degree order $>_s$ on $M$, where we break ties by declaring the monomial with $z$ precedes the other without $z$. For $f \in Rz \oplus \bar{W}$, the notations $\text{l}_s(f)$, $\text{lm}_s(f)$, and $\text{lc}_s(f)$ are used to denote respectively the leading term, the leading monomial, and the leading coefficient, with respect to $>_s$. Suppose $M$ is an $F[x]$-submodule of $Rz \oplus \bar{W}$. A finite subset $B$ of $M$ is called a Gröbner basis of $M$ with respect to $>_s$ if the leading term of every element of $M$ is an $F[x]$-multiple of the leading term of some element of $B$. We will write

$B = \{g_i, f_i : 0 \leq i < \gamma\}$,

where we understand that the leading term of $g_i$ is in $\bar{W}$ while that of $f_i$ is in $Rz$. The sigma set $\Sigma_s = \Sigma_s(M)$ of $M$ is the set of all leading monomials of the
polynomials in $M$ with respect to $>_s$. The delta set $\Delta_s = \Delta_s(M)$ of $M$ is the complement of $\Sigma_s$ in $M$.

The residue map

$$\text{res} : \overline{W} \rightarrow \mathbb{F}^n, \quad \omega \mapsto (\text{res}_{p_1}(\omega), \text{res}_{p_2}(\omega), \ldots, \text{res}_{p_n}(\omega))$$

is linear over $\mathbb{F}$. Thus the differential AG code $C = C_{\Omega}(D, G) = \text{res}(\Omega(-D + G))$ is a linear code of length $n$ over $\mathbb{F}$. Note that $\{\tilde{\phi}_s : s \in \tilde{\Omega}, s \leq 0\}$ is a basis of $\Omega(-D + G)$ as a vector space over $\mathbb{F}$. Let $k$ be the dimension of $C$. Then there is a set $S = \{s_0, s_1, \ldots, s_{k-1}\} \subset \{s : s \in \tilde{\Omega}, s \leq 0\}$ such that $\{\text{res}(\tilde{\phi}_s) : s \in S\}$ is a basis of $C$. Note that the map res is surjective onto $\mathbb{F}^n$. Indeed we can show that $\text{res}(\Omega(-D + G - sQ)) = \mathbb{F}^n$ for $s > |G| = \deg(G)$ by the Riemann-Roch theorem. Let $h_i \in \overline{W}$ be the differential such that $\text{res}(h_i)$ is the $i$th element of the standard basis of $\mathbb{F}^n$. Let $J$ be the kernel of res. Note that $J$ is a submodule of $\overline{W}$ over $\mathbb{R}$, and also over $\mathbb{F}[x]$. Let $\{\eta_i : 0 \leq i < \gamma\}$ be a Gröbner basis of $J$ over $\mathbb{F}[x]$ such that $\deg_s(\text{lt}(\eta_i)) = i$. Then we have

$$\sum_{0 \leq i < \gamma} \deg_s(\text{lt}(\eta_i)) = |\Delta(J)| = \dim \overline{W}/J = n,$$

which corresponds to Proposition 2 of [10] and can be proved in a similar way.

3. DECODING ALGORITHM

We assume a codeword is sent through a communication channel and a vector $v \in \mathbb{F}^n$ is received. Thus we suppose $v = c + e$ with a codeword $c$ and the error vector $e$. Then $c = \text{res}(\mu)$ with a unique differential

$$(2) \quad \mu = \sum_{s \in S} m_s \tilde{\phi}_s \in \Omega(-D + G).$$

For all integer $0 < s \in \tilde{\Omega}$, let $v^{(s)} = v$, $c^{(s)} = c$, and $\mu^{(s)} = \mu$. For $s \in S$, let

$$\mu^{(s-1)} = \mu^{(s)} - m_s \tilde{\phi}_s, \quad c^{(s-1)} = c^{(s)} - \text{res}(m_s \tilde{\phi}_s),$$

$$v^{(s-1)} = v^{(s)} - \text{res}(m_s \tilde{\phi}_s),$$

and for $s \leq 0$ but $s \notin S$, let $\mu^{(s-1)} = \mu^{(s)}$, $c^{(s-1)} = c^{(s)}$, $v^{(s-1)} = v^{(s)}$. Then we have $\mu^{(s)} \in \Omega(-D + G - sQ)$, $c^{(s)} = \text{res}(\mu^{(s)})$, and $v^{(s)} = c^{(s)} + e$ for all $s$.

The interpolation module $I_v$ for a vector $v = (v_1, v_2, \ldots, v_n)$ is now defined by

$$(3) \quad I_v = \{fz + \omega \in Rz \oplus \overline{W} : f(P_i)v_i + \text{res}_{p_i}(\omega) = 0, 1 \leq i \leq n\},$$

which is a submodule of $Rz \oplus \overline{W}$ over $R$. Let $h_v = \sum_{i=1}^n v_i h_i$ so that $\text{res}(h_v) = v$. Then we see that $I_v = R(z - h_v) + J$. Hence the set

$$(4) \quad \{\eta_i, y_i(z - h_v) : 0 \leq i < \gamma\}$$

is a Gröbner basis of $I_v$ with respect to $>_D(h_v)$.

The Fast Decoding Algorithm for differential AG codes, displayed in Figure 1, starts with the basis (4) and iterates the substeps Pairing, Voting, and Rebasing, computing a Gröbner basis of $I_{v^{(s-1)}}$ from that of $I_{v^{(s)}}$ while $m_s \in \mathbb{F}$ is computed by majority voting for $s \in S$. Let

$$v^{(s)} = \frac{1}{\gamma} \sum_{0 \leq i < \gamma} \max\{\delta(\eta_i^c) - \rho(y_i) - s, 0\} = |\Delta(J) \cap \Sigma(R \tilde{\phi}_s)|$$
for $s \in S$, and define $d_\Omega = \min\{\nu(s) : s \in S\}$. Then it can be shown that the value $m_s$, computed by the algorithm equals to the coefficient $m_s$ of $\mu$ in (2), if $2\text{wt}(e) < \nu(s)$ for $s \in S$. Hence the algorithm succeeds in iteratively computing $m_s$ for all $s \in S$ if $\text{wt}(e) \leq \tau = [(d_\Omega - 1)/2]$. The proof is mostly identical with the corresponding one in [10], with some obvious change of notations. So we leave out the proof.

Note that the Fast Decoding Algorithm for differential AG codes is also enhanced with the speedup techniques introduced in the Section III.E of [10] for evaluation AG codes. A minor difference is due to the different inequality in the following theorem.

**Theorem 3.1.** Suppose $\mu \in \Omega(-D + G - sQ)$ and $v \in \mathbb{F}^n$. If for a nonzero $f \in I_v$,
\[
\delta_s(f) + \text{wt}(v - \text{res}(\mu)) + 2g - 1 \leq |G|,
\]
then $f(\mu) = 0$. Here $f(\mu)$ denotes the $f$ with the variable $z$ substituted with $\mu$.

**Proof.** Let $\delta_s(f) = d$. Then $f(\mu) \in \Omega(-D + G - dQ)$. As $f \in I_v$, $\text{res}_{P_i}(f(\mu)) = 0$ for $i$ with $v_i = \text{res}_{P_i}(\mu)$. It follows that
\[
f(\mu) \in \Omega \left( - \sum_{v_i \neq \text{res}_{P_i}(\mu)} P_i + G - dQ \right).
\]
Hence by the Riemann-Roch theorem, if $f(\mu)$ is nonzero, then we must have $-\text{wt}(v - \text{res}(\mu)) + |G| - d < 2g - 1$. \qed

According to Theorem 3.1, a polynomial $f$ in $I_v(e)$ satisfying the condition $\delta_s(f) + \text{wt}(e) + 2g - 1 \leq |G|$ is called a $Q$-polynomial for $v(e)$. In the Fast Decoding Algorithm, we actually use the condition $\delta_s(f) + \tau + 2g - 1 \leq |G|$ since if we assume $\text{wt}(e) \leq \tau$, then a polynomial $f$ in $B(e)$ satisfying the condition is a $Q$-polynomial for $v(e)$. See the step Q. Since $|\Delta_s(I_v(e)) \cap Rz| \leq \text{wt}(e)$ (see Corollary 7 in [10]), if we have $|\Delta_s(I_v(e)) \cap Rz| > \tau$ during decoding, then we must have $\text{wt}(e) > \tau$, and the decoder may declare Decoding Failure. See the step F.

### 4. Performance analysis

#### 4.1. Decoding capacity

**Proposition 1.** We have $\nu(s) = |\Delta(J) \cup \Delta(R\tilde{z}_s)| - n + |G| - 2g + 2 - s$. Hence $d_\Omega$ is at least Goppa bound $|G| - 2g + 2$ for differential AG codes.

**Proof.** Note that
\[
\nu(s) = |\Delta(J) \cap \Sigma(R\tilde{z}_s)|
\]
\[
= |\Delta(J) \cup \Delta(R\tilde{z}_s)| - |\Delta(R\tilde{z}_s)|
\]
\[
= |\Delta(J) \cup \Delta(R\tilde{z}_s)| - n + |G| - s - 2g + 2.
\]
To see the last equality, note that by the Riemann-Roch theorem,
\[
|\Delta(R\tilde{z}_s)| = \dim \mathbb{W}/R\tilde{z}_s = \dim \Omega(-D + G - (t+s)Q)/L(tQ)\tilde{z}_s = n - |G| + s + 2g - 2
\]
for all large enough $t$. Since $|\Delta(J) \cup \Delta(R\tilde{z}_s)| \geq n$, we have $\nu(s) \geq |G| - 2g + 2 - s$. Now our assertion follows. \qed
Fast Decoding Algorithm. Let $v \in \mathbb{F}^n$ be the received vector.

**S1** Compute $h_v = \sum_{i=1}^{n} v_i h_i$. Let $N = \delta(h_v)$, and set

$$B^N = \{\eta_v, y_i(z - h_v) : 0 \leq i < \gamma\}.$$  

Let $m_s = 0$ for $s$ with $N < s \in S$. If $N \leq 0$, then set $m_s \in \mathbb{F}$ such that $h_v = \sum_{s \in S} m_s \varphi_s$, and go to the step S3.

**S2** Repeat the following for $s$ from $N$ to $0$. Let $B_i = \{g_i^{(s)}, f_i^{(s)} : 0 \leq i < \gamma\}$ be a Gröbner basis of $I_{v(i)}$ with respect to $\succ_s$ where

$$g_i^{(s)} = \sum_{0 \leq j < \gamma} c_{i,j} y_j z + \sum_{0 \leq j < \gamma} d_{i,j} \omega_j$$

$$f_i^{(s)} = \sum_{0 \leq j < \gamma} a_{i,j} y_j z + \sum_{0 \leq j < \gamma} b_{i,j} \omega_j$$

and let $\nu_i^{(s)} = \text{lc}(d_{i,i})$.

**F** If $\sum_{0 \leq i < \gamma} \deg(a_{i,i}) > \tau$, then declare Decoding Failure and stop.

**Q** If there is an $i$ such that $\gamma \deg(a_{i,i}) + a_i + \tau + 2g - 1 \leq |G|$, then set $f_i^{(s)} = \varphi z + \psi$. Set $m_t \in \mathbb{F}$ such that $- \varphi^{-1} \psi = \sum_{i \in S, t \leq s} m_t \omega_i$, and proceed to the step S3. If $\psi$ is not divisible by $\varphi$, then declare Decoding Failure and stop.

**M Pairing.** For $0 \leq i < \gamma$, let $i' = (i + s) \mod \gamma$, $k_i = \deg(a_{i,i}) + (a_i + s - b_{i'})/\gamma$, and $c_i = \deg(d_{i,i'}) - k_i$.

**Voting.** If $s \in S$, then for each $i$, let $m_i = - \frac{b_{i,i'}}{\mu_i} \varphi_{i'}$, $\mu_i = \text{lcm}(a_{i,i}, y_i \varphi_s)$ and let $\bar{c}_i = \max\{c_i, 0\}$, and let $m = \text{the element of } \mathbb{F}$ with the largest $\sum_{i=m} \bar{c}_i$, and let $m_s = m$. If $s \notin S$, then for $i$ with $k_i \geq 0$, let $m_i = - \frac{b_{i,i'}}{\mu_i} , \mu_i = 1$ and for $i$ with $k_i < 0$, let $m_i = 0, \mu_i = 1$. Let $m = 0$ in both cases.

**Rebasings.** For each $i$, do the following. If $m_i = m$, then let

$$g_i^{(s-1)} = g_i^{(s)}(z + m \varphi_s), \quad f_i^{(s-1)} = f_i^{(s)}(z + m \varphi_s)$$

and let $\nu_i^{(s-1)} = \nu_i^{(s)}$. If $m_i \neq m$ and $c_i > 0$, then let

$$g_i^{(s-1)} = f_i^{(s)}(z + m \varphi_s)$$

$$f_i^{(s-1)} = x^{c_i} f_i^{(s)}(z + m \varphi_s) - \frac{\mu_i (m - m_i)}{\nu_i^{(s)}} g_i^{(s)}(z + m \varphi_s)$$

and let $\nu_i^{(s-1)} = \mu_i (m - m_i)$. If $m_i \neq m$ and $c_i \leq 0$, then let

$$g_i^{(s-1)} = g_i^{(s)}(z + m \varphi_s)$$

$$f_i^{(s-1)} = f_i^{(s)}(z + m \varphi_s) - \frac{\mu_i (m - m_i)}{\nu_i^{(s)}} x^{-c_i} g_i^{(s)}(z + m \varphi_s)$$

and let $\nu_i^{(s-1)} = \nu_i^{(s)}$. Let $B^{(s-1)} = \{g_i^{(s-1)}, f_i^{(s-1)} : 0 \leq i < \gamma\}$.

**S3** Output the codeword $\sum_{s \in S} m_s \text{res} (\varphi_s)$.

**Figure 1.** Fast decoding algorithm for differential AG codes

---

**Figure 1.** Fast decoding algorithm for differential AG codes

---

**Figure 1.** Fast decoding algorithm for differential AG codes
Decoding of differential AG codes

Let us define $\tau_M(s) = \lfloor (\nu(s)-1)/2 \rfloor$ for $s \in \Omega$, which is the largest number of errors for which the majority voting succeeds for $s$. Like Proposition 22 in [10], we can show that for nongap $s \leq |G| - 2g + 2$,
\begin{equation}
(5) \quad \lfloor (|G| - 2g + 1 - s)/2 \rfloor \leq \tau_M(s) \leq \lfloor (|G| - g + 1 - s)/2 \rfloor
\end{equation}
and if $s \leq |G| - 4g + 2$, equality holds on the left.

We now find out when the condition in the step Q is satisfied. Let $t = \text{wt}(e)$.

**Theorem 4.1.** Let $B^{(s)}$ be a Gröbner basis of $I_{v^{(s)}}$ with respect to $\succ_s$. If $\lambda_t + s + t + 2g - 1 \leq |G|$, then there exists an $f \in B^{(s)}$ such that $f$ is a $Q$-polynomial for $v^{(s)}$.

**Proof.** By the same argument of the proof of Lemma 18 of [10], we know that there exists an $f \in B^{(s)}$ such that $\delta_s(f) \leq \lambda_t + s$. So if $\lambda_t + s + t + 2g - 1 \leq |G|$, then this $f$ satisfies the condition to be a $Q$-polynomial. \qed

Let us define
\[
\tau_Q(s) = \max \{ t : \lambda_t + s + t + 2g - 1 \leq |G| \}
\]
for $s \in \Omega$. By Theorem 4.1, the value $\tau_Q(s)$ is the largest number of errors for which a $Q$-polynomial exists in $B^{(s)}$. Like Proposition 20 in [10], we can then show that for $s \in \Omega$,
\begin{equation}
(6) \quad \lfloor (|G| - 3g + 1 - s)/2 \rfloor \leq \tau_Q(s) \leq \lfloor (|G| - 2g + 1 - s)/2 \rfloor
\end{equation}
and for $s \leq |G| - 5g + 1$, equality holds on the left. From (5) and (6), we see that $\tau_Q(s) \leq \tau_M(s)$ for nongap $s \leq |G| - 2g + 2$, and moreover for $s \leq |G| - 5g + 1$,
\[
\tau_M(s) - \tau_Q(s) = \begin{cases} 
\lfloor g/2 \rfloor & \text{if } |G| - s \text{ is odd}, \\
\lfloor g/2 \rfloor & \text{if } |G| - s \text{ is even}.
\end{cases}
\]

Recall that the actual condition used in the step Q to find a $Q$-polynomial in $B^{(s)}$ is $\delta_s(f) + \tau + 2g - 1 \leq |G|$. By a similar proof of Theorem 4.1, we have

**Theorem 4.2.** Suppose $t = \text{wt}(e) \leq \tau$. If $\lambda_t + s + \tau + 2g - 1 \leq |G|$, then there exists an $f \in B^{(s)}$ satisfying $\delta_s(f) + \tau + 2g - 1 \leq |G|$, and $f$ is a $Q$-polynomial for $v^{(s)}$.

By Theorem 4.2, the condition in the step Q is satisfied for some $s \geq s_Q(t) = |G| - \lambda_t - \tau - 2g + 1$ depending on $t = \text{wt}(e)$, and at the latest for some $s \geq s_Q(\tau) = |G| - \lambda_\tau - \tau - 2g + 1$. Finally note that
\begin{equation}
(7) \quad \tau = \min_{s \in S} \tau_M(s), \quad s_Q(\tau) = \max \{ s : \tau_M(s) \geq \tau \},
\end{equation}
which can be verified by definitions.

4.2. **Complexity.** The Fast Decoding Algorithm iteratively updates a $2\gamma \times 2\gamma$ array of polynomials in $\mathbb{F}[x]$ that represents $B^{(s)}$. Each of the $2\gamma$ rows of the array are again viewed as pairs of vectors in $\mathbb{F}[x]^\gamma$. For the initialization step S1, we precompute $h_i$ for $1 \leq i \leq n$ and $\eta_i$ for $0 \leq i < \gamma$ in the vector form. In the **Rebasing** substep of the step M, the most intensive computation is the substitution of $z$ with $z + m\tilde{\omega}_i$. As $\tilde{\omega}_i$ is in the form $x^k\tilde{\omega}_i$, the computation is facilitated if $\eta_i \tilde{\omega}_j$ for $0 \leq i, j < \gamma$ is precomputed in the vector form. For the step S3, it is necessary to precompute the vectors $\text{res}(\tilde{\omega}_{s_i})$ in $\mathbb{F}^n$ for $0 \leq i \leq k - 1$, essentially the generator matrix of the code $C$. Our complexity analysis is now summarized, omitting the details, in the following.
Proposition 2. (1) Lagrange basis polynomial $h_i$ can be chosen such that the maximum degree of the polynomials in the vector form of $h_i$ is bounded by $N_h = [(n + 2g - 1)/\gamma]$.

(2) The maximum degree of the polynomials in the vector form of $\eta_i$ is bounded by $N_{\eta} = [(n + 3g + \gamma - 1)/\gamma]$.

(3) The maximum degree of the polynomials in the $2\gamma \times 2\gamma$ array during an execution is bounded by $N_{\text{deg}} = 1 + [(n + 4g - 2)/\gamma]$ if $g > 0$. If $g = 0$, then it is bounded by $n$.

(4) The number of iterations is at most $N_{\text{iter}} = n + 2g$.

Proposition 3. If $g > 0$, an execution of the Fast Decoding Algorithm takes $O((n + 4g + \gamma)(2\gamma + 3g)(2g + \gamma)\gamma)$ multiplications. For $g = 0$, it takes $O(n^2)$ multiplications. The implicit constant is absolute.

Observe that these results are exactly the same with the complexity analysis of the decoding algorithm for evaluation AG codes reported in [10].

4.3. Comparisons of minimum distance bounds. We now show that $d_\Omega$ is indeed a lower bound for the minimum distance of the code $C_{\Omega}(D, G)$. Recall that $d_\Omega = \min\{\nu(s) : s \in S\}$ where $\nu(s) = |\Delta(J) \cap \Sigma(R\bar{\varphi}_s)| = n - |\Delta(J) \cap \Delta(R\bar{\varphi}_s)|$.

Theorem 4.3. The minimum distance $d$ of $C_{\Omega}(D, G)$ is lower-bounded by $d_\Omega$.

Proof. Let $s \in S$ and suppose $c = \text{res}(\mu) = \sum_{i \leq s} a_i \hat{\varphi}_i \in \Omega(-D + G)$ with nonzero $a_i \in \mathbb{F}$. Consider $\mathbb{F}^n$ as an $\mathbb{F}$-algebra with the component-wise multiplication $\ast$. Let us consider the evaluation map from $R$ to $\mathbb{F}^n$, which is a surjective homomorphism of $\mathbb{F}$-algebras. Let $J$ be the kernel of the map, and we have an isomorphism $R/J$ with $\mathbb{F}^n$. Then

$$n - \text{wt}(c) = \dim_{\mathbb{F}} \{v \in \mathbb{F}^n : v \ast c = 0\}$$

$$= \dim_{\mathbb{F}} \{f \in R/J : \text{res}(f \mu) = 0\}$$

$$= \dim_{\mathbb{F}} \{f \in R/J : f \mu \in J\}$$

$$= \dim_{\mathbb{F}} \ker(R/J \twoheadrightarrow W/J)$$

$$= n - \dim_{\mathbb{F}} \text{im}(R/J \twoheadrightarrow W/J)$$

$$= n - \dim_{\mathbb{F}} (J + R\mu)/J$$

$$= \dim_{\mathbb{F}} W/(J + R\mu)$$

$$= |\Delta(J + R\mu)|$$

Hence $\text{wt}(c) = n - |\Delta(J + R\mu)|$. Now as $\Sigma(R\mu) = \Sigma(R\bar{\varphi}_s)$, we have $|\Delta(J + R\mu)| \leq |\Delta(J) \cap \Delta(R\bar{\varphi}_s)|$. Therefore $\nu(s) \geq n - |\Delta(J) \cap \Delta(R\bar{\varphi}_s)| = \nu(s) \geq d_\Omega$. This result implies that $d \geq d_\Omega$. \qed

Recall that

$$\Lambda = \{i : L(iQ) \neq L(i-1)Q\},$$

$$\bar{\Lambda} = \{s : L(G + sQ) \neq L(G + (s - 1)Q)\},$$

$$\bar{\Omega} = \{s : \Omega(-D + G - sQ) \neq \Omega(-D + G - (s - 1)Q)\},$$

where $\bar{\Lambda}$ is the numerical $\Lambda$-module used in [10]. Then Theorem 33 in [10] says that the minimum distance of $C_\mathcal{C}(D, G)$ is lower bounded by

$$d_\mathcal{C} = \min_{s \in \bar{\Lambda}, s \leq 0} \left|\{(i, j) : i \in \Lambda, j \in \bar{\Omega}, i + j + s = 1\}\right|.$$
For comparison of $d_\Omega$ with $d_\mathcal{L}$, we assume $\deg(G) \geq 2g - 1$ such that $S = \{s \in \Lambda, s \leq 0\}$ from now on. Like Lemma 32 of [10], we can show that $a + 1 \in \Lambda$ if and only if $-a \notin \Omega$ or $\varphi_{-a} \in \Delta(J)$, which implies that

$$d_\Omega = \min_{s \in \Omega, s \leq 0} |\{(i, j) : i \in \Lambda, j \in \bar{\Lambda}, i + j + s = 1\}|.$$

Observe the nice symmetry between $d_\mathcal{L}$ and $d_\Omega$.

Recall that $ev(f) = (f(P_1), f(P_2), \ldots, f(P_n))$ for $f \in \mathcal{R} = \bigcup_s \mathcal{L}(G + sQ)$. Let $C_i = ev(\mathcal{L}(G+iQ)) \subset \mathbb{F}^n$ for $i \in \mathbb{Z}$. The codes $C_i$ are sometimes used in formulating minimum distance bounds of AG codes (see [6, 5]).

**Lemma 4.4.** $s \in \bar{\Omega}$ if and only if $-s + 1 \notin \bar{\Lambda}$ or $C_{-s} \neq C_{-s+1}$.

**Proof.** By the Riemann-Roch theorem,

$$s \in \bar{\Omega} \iff \dim_\mathbb{F} \Omega(-D + G + (-s + 1)Q) \neq \dim_\mathbb{F} \Omega(-D + G - sQ)$$

$$\iff \dim_\mathbb{F} \mathcal{L}(-D + G + (-s + 1)Q) = \dim_\mathbb{F} \mathcal{L}(-D + G - sQ)$$

$$\iff -s + 1 \notin \bar{\Lambda} \text{ or } C_{-s+1} \neq C_{-s}$$

For the last equivalence, recall that $\dim_\mathbb{F} C_i = \dim_\mathbb{F} \mathcal{L}(G + iQ) - \dim_\mathbb{F} \mathcal{L}(D + G + iQ)$.

Lemma 4.4 allows us to write

$$d_\Omega = \min_{C_k \neq C_{k-1}, k \geq 1} |\{(i, j) : i \in \Lambda, j \in \bar{\Lambda}, i + j = k\}|,$$

$$d_\mathcal{L} = \min_{s \in \Lambda, s \leq 0} |\{i \in \Lambda : C_{i+s} \neq C_{i+s-1}\}|.$$

Finally let us focus on one-point AG codes with $G = mQ$. In this case $\bar{\Lambda} = \Lambda - m$. Let $H_i = ev(\mathcal{L}(iQ)) = C_{i-m}$ and $H^* = \{i : H_i \neq H_{i-1}\} \subset \Lambda$. Then

$$d_\Omega = \min_{k \in H^*, k > m} |\{(i, j) : i \in \Lambda, i + j = k\}|$$

$$\geq \min_{k \in H^*, k > m} |\{(i, j) : i \in H^*, i + j = k\}|,$$

$$d_\mathcal{L} = \min_{s \in \Lambda, s \leq m} |\{i \in \Lambda : i + s \in H^*\}|$$

$$\geq \min_{s \in \Lambda, s \leq m} |\{i \in H^* : i + s \in H^*\}|,$$

where the former is the famous Feng-Rao bound of the one-point AG code $C_\Omega(D, mQ)$ and the latter is a slight variation of the bound $d^*$ defined in [6].

**5. Example**

Let $X$ be the Hermitian curve of genus $g = 3$ defined by the equation

$$y^3 + y = x^4$$

over $\mathbb{F}_9 = \mathbb{F}_3(\alpha)$ with $\alpha^2 - \alpha - 1 = 0$. Let $G = -O + 18Q$ where $O$ is the origin and $Q$ is the unique point at infinity. Except $O$ and $Q$, there are 26 rational points

$$(0, \alpha^2), (0, \alpha^6), (1, 2), (1, \alpha), (1, \alpha^3), (2, 2), (2, \alpha),$$

$$(2, \alpha^3), (\alpha, 1), (\alpha, \alpha^7), (\alpha, \alpha^5), (\alpha^2, 2), (\alpha^2, \alpha), (\alpha^2, \alpha^3),$$

$$(\alpha^7, 1), (\alpha^7, \alpha^7), (\alpha^7, \alpha^5), (\alpha^9, 1), (\alpha^9, \alpha^7), (\alpha^9, \alpha^5),$$

$$(\alpha^3, 1), (\alpha^3, \alpha^7), (\alpha^3, \alpha^5), (\alpha^6, 2), (\alpha^6, \alpha), (\alpha^6, \alpha^3).$$
Then the differential AG code $C = C_{12}(D, G)$ is a $[26, 11, 13]$ linear code over $\mathbb{F}_9$. Note that this is the dual code of $C_{12}(D, G)$ dealt in Section IV-A of [10]. Hence the data about $R$ are the same, but the data about $\overline{W}$ are new.

The Weierstrass semigroup at $Q$ is
\[ \Lambda = \{0, 3, 4, 6, 7, 8, 9, \ldots \}. \]
So $\gamma = 3$, and we take $x = x$. The Apéry system of $R$ is
\[
\begin{align*}
y_0 &= 1, & \rho(y_0) &= 0, \\
y_1 &= y, & \rho(y_1) &= 4, \\
y_2 &= y^2, & \rho(y_2) &= 8.
\end{align*}
\]
On the other hand,
\[ \bar{\Omega} = \{-13, -10, -9, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, \ldots \}, \]
and the Apéry system of $\bar{W}$ is
\[
\begin{align*}
\bar{\omega}_0 &= y(x^9 - x)^{-1}dx, & \delta(\bar{\omega}_0) &= -9, \\
\bar{\omega}_1 &= y^2(x^9 - x)^{-1}dx, & \delta(\bar{\omega}_1) &= -5, \\
\bar{\omega}_2 &= (x^9 - x)^{-1}dx, & \delta(\bar{\omega}_2) &= -13.
\end{align*}
\]
The monomials of $R_2 \oplus \overline{W}$ are displayed in the array below, where the common $z$ factor of the monomials of $R_2$ are omitted.

\[
\begin{array}{cccccccccccccccc}
\omega_0 & 1 & x & x^2 & x^3 & x^4 & x^5 & x^6 & x^7 & x^8 & x^9 \\
\omega_1 & 1 & x & x^2 & x^3 & x^4 & x^5 & x^6 & x^7 & x^8 & x^9 \\
\omega_2 & 1 & x & x^2 & x^3 & x^4 & x^5 & x^6 & x^7 & x^8 & x^9
\end{array}
\]
Here are the corresponding nongaps of $\Lambda$ and $\bar{W}$, respectively.

\[
\begin{array}{cccccccccccccccc}
8 & 11 & 14 & 17 & 20 & 23 & 26 & 29 & 32 & 35 & \\
4 & 7 & 10 & 13 & 16 & 19 & 22 & 25 & 28 & 31 & \\
0 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 & \\
-13 & -10 & -7 & -4 & -1 & 2 & 5 & 8 & 11 & 14 & \\
-5 & -2 & 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 & \\
-9 & -6 & -3 & 0 & 3 & 6 & 9 & 12 & 15 & 18 & 
\end{array}
\]
The Lagrange interpolation polynomials in $\bar{W}$ are
\[
\begin{align*}
h_1 &= (-x^8 + 1)\bar{\omega}_2 + (\alpha^2 x^8 + \alpha^6)\bar{\omega}_0, \\
h_2 &= (-x^8 + 1)\bar{\omega}_2 + (\alpha^6 x^8 + \alpha^2)\bar{\omega}_0, \\
\vdots
\end{align*}
\]
\[
\begin{align*}
h_{26} &= (\alpha^6 x^8 + \alpha^3 x^7 + \cdots + 1)\bar{\omega}_2 \\
&+ (\alpha^6 x^7 + 2x^6 + \cdots + 1)\bar{\omega}_1 \\
&+ (\alpha^3 x^8 + \alpha x^7 + \cdots + \alpha^5 x)\bar{\omega}_0.
\end{align*}
\]
The $\mathbb{F}_9[x]$-submodule $J$ of $\overline{W}$ has Gröbner basis $\{\eta_0, \eta_1, \eta_2\}$, where
\[
\begin{align*}
\eta_0 &= (x^9 - x)\bar{\omega}_0, \\
\eta_1 &= (x^9 - 1)\bar{\omega}_1 + (x^8 - 1)\bar{\omega}_2, \\
\eta_2 &= (x^9 - x)\bar{\omega}_2.
\end{align*}
\]
Using these data, we can compute

| \(s\) | \(\nu(s)\) |
|---|---|
| 0  | 13  |
| -6 | 19  |
| -1 | 14  |
| -7 | 20  |
| -2 | 15  |
| -9 | 22  |
| -3 | 16  |
| -10| 23  |
| -4 | 17  |
| -13| 26  |
| -5 | 18  |

and hence \(d_0 = 13\). The Fast Decoding Algorithm runs with the received vector \(v\) as input and with the above precomputed data. The decoding process itself is similar to that of the example in Section IV-A in [10]. Thus we omit an example run of the algorithm.

### 6. Decoding Goppa codes

Let \(L = \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \subseteq F\) be a set of distinct rational points of the projective line \(P_F\) over \(F = F_q^m\), and let \(P_\infty\) denote the point at infinity. Let \(D\) denote the divisor of zeros of \(\prod_{k=1}^{n}(x - \alpha_k)\). Let \(g(x) \in F[x]\) be a given polynomial with \(g(\alpha_i) \neq 0\) for \(1 \leq i \leq n\), and let \(Z\) be the divisor of zeros of \(g(x)\). We assume \(\deg g(x) < n\). Then the classical Goppa code \(C|_{F_q}\) is a subfield subcode over \(F_q\) of the differential AG code

\[ C = C_0(D, Z - P_\infty). \]

As \(C\) is an \([n, k, d]\) code with \(k = n - \deg g(x)\) and \(d \geq \deg g(x) + 1\), the Goppa code \(C|_{F_q}\) is an \([n, k', d']\) linear code over \(F_q\) with \(k' \geq n - m \deg g(x), d' \geq \deg g(x) + 1\). See [12] for more on Goppa codes and subfield subcodes of AG codes.

Clearly, the Fast Decoding Algorithm can decode the subfield subcode \(C|_{F_q}\) just by decoding \(C\). So let us specialise the algorithm for \(C\). Let \(X = P_F\). The genus of the projective line is \(g = 0\), and \(F(X) = F(x), R = F[x], \Lambda = \mathbb{Z}_{\geq 0}\). For \(f(x) \in F[x]\), simply \(\rho(f(x)) = \deg f(x)\). In particular, \(\rho(x) = \gamma = 1\) and \(\rho(y_0) = a_0 = 0\) with \(y_0 = 1\). Furthermore \(\Omega_X = F(x)dx\) and we can show that \(\bar{W} = F[x]\bar{\omega}_0\) and \(\bar{\Omega} = \mathbb{Z}_{\geq 0} + b_0\) where

\[
\bar{\omega}_0 = \frac{g(x)}{\prod_{i=1}^{n}(x - \alpha_i)}dx, \quad b_0 = \deg g(x) - n + 1,
\]

noting that \(\delta(f(x)\bar{\omega}_0) = \deg f(x) + b_0\). For \(\omega = f(x)\bar{\omega}_0 \in \bar{W}\), we have

\[
\text{res}(\omega) = (f(\alpha_1)g(\alpha_1), f(\alpha_2)g(\alpha_2), \ldots, f(\alpha_n)g(\alpha_n))
\]

Therefore \(J = F[x]\eta_0\) where

\[
\eta_0 = \prod_{i=1}^{n}(x - \alpha_i)\bar{\omega}_0
\]

and for \(1 \leq i \leq n\),

\[
h_i = \frac{\prod_{j=1,j \neq i}^{n}(x - \alpha_j)}{g(\alpha_i)\prod_{j=1,j \neq i}^{n}(\alpha_i - \alpha_j)}\bar{\omega}_0.
\]

With the above precomputed data, the Fast Decoding Algorithm for Goppa codes is reduced to the following simple algorithm.

Note that the differential \(\bar{\omega}_0\), as well as \(z\), is just a placeholder, and the algorithm actually works with a \(2 \times 2\) array of univariate polynomials. We made some changes in notations for clarification and restructured the algorithm slightly for
read implementation. It should be noted that when the algorithm is in second phase \( s < n - \deg g(x) \), the update of \( G^{(s)} \) is actually unnecessary.

**Fast Decoding Algorithm for Goppa Codes.** Let \( v \in \mathbb{F}^n \) be the received vector.

**S1** Compute \( h_v = \sum_{i=1}^{n} v_i h_i \). Let \( G^{(n-1)} = \eta_0 \), \( F^{(n-1)} = z - h_v \) and \( \nu^{(n-1)} = 1 \).

**S2** Repeat the following for \( s \) from \( n - 1 \) to 0.

- **M** Suppose \( G^{(s)} = Cz + D\omega_0 \), \( F^{(s)} = Az + B\omega_0 \). Let \( k = \deg(A) + s \), and \( c = \deg(D) - k \).
  - If \( s \geq n - \deg g(x) \), then let \( m = -B[x^k] \). If \( m = 0 \), let \( G^{(s-1)} = G^{(s)} \), \( F^{(s-1)} = F^{(s)} \), and \( \nu^{(s-1)} = \nu^{(s)} \).
  - If \( m \neq 0 \) and \( c > 0 \), then let \( G^{(s-1)} = F^{(s)} \), \( F^{(s-1)} = \nu^{(s)} x^c F^{(s)} + m G^{(s)} \), and let \( \nu^{(s-1)} = -m \).
  - If \( m \neq 0 \) and \( c \leq 0 \), then let \( G^{(s-1)} = G^{(s)} \), \( F^{(s-1)} = \nu^{(s)} F^{(s)} + m x^{-c} G^{(s)} \), and let \( \nu^{(s-1)} = \nu^{(s)} \).

- If \( s < n - \deg g(x) \), then let \( m_s = -\text{lc}(A)^{-1} B[x^k] \) and let \( G^{(s-1)} = G^{(s)} (z + m_s x^s \omega_0) \), \( F^{(s-1)} = F^{(s)} (z + m_s x^s \omega_0) \), and let \( \nu^{(s-1)} = \nu^{(s)} \).

**S3** Let \( \mu = \sum_{s=0}^{n-\deg g(x)-1} m_s x^s \) and \( r = (\mu(\alpha_i)g(\alpha_i) : 1 \leq i \leq n) \). If \( r \in \mathbb{F}^n \), output \( r \). Otherwise declare **Decoding Failure**.

As is well-known, the binary Goppa code \( C_{\mathbb{F}_2} \) defined by a separable polynomial \( g(x) \in \mathbb{F}_{2^n}[x] \) is identical with the Goppa code defined by \( g(x)^2 \). Hence \( C_{\mathbb{F}_2} \) has dimension \( \geq n - m \deg g(x) \) and minimum distance \( 2 \deg g(x) + 1 \). Moreover the Goppa code \( C_{\mathbb{F}_2} \) can be decoded up to \( \deg g(x) \) number of errors by the Fast Decoding Algorithm for the Goppa code defined by \( g(x)^2 \). We take this approach in the following example.

**Example.** Let \( C_{\mathbb{F}_2} \) be the binary Goppa code defined by

\[
L = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^6\} \subset \mathbb{F}_8 = \mathbb{F}_2[\alpha]
\]

with \( \alpha^3 + \alpha + 1 = 0 \) and the generator polynomial \( g(x)^2 \in \mathbb{F}_8[x] \) with \( g(x) = z^2 + z + 1 \). This is an \([8, 2, 5]\) Goppa code, capable of correcting two errors. Precomputed are \( \eta_0 = (x^8 + x)\omega_0 \) and

\[
h_1 = (x^7 + 1)\omega_0,
\]

\[
h_2 = (x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x)\omega_0,
\]

\[
\vdots
\]

\[
h_8 = (\alpha x^7 + x^6 + \alpha^6 x^5 + \alpha^5 x^4 + \alpha^4 x^3 + \alpha^3 x^2 + \alpha^2 x)\omega_0
\]

Suppose \( v = (1, 1, 1, 1, 1, 1, 0) \) is the received vector. The algorithm iteratively computes the following for \( s = 7, 6, \ldots, 0 \).

\[
G^{(s)} = \begin{cases} \sum_{i=1}^{8} v_i h_i \end{cases} \quad F^{(s)} = \begin{cases} \sum_{i=1}^{8} v_i h_i \end{cases}
\]

\[
F^{(7)} = 1 z + (\alpha x^7 + \alpha^2 x^5 + \alpha^4 x^4 + \alpha^3 x^3 + \alpha^2 x^2 + \alpha x + 1)\omega_0
\]
Decoding of differential AG codes

$$G^{(6)} = 0z + (\alpha x^7 + \alpha^2 x^5 + \alpha^3 x^4 + \alpha^5 x^3 + \alpha^3 x^2 + \alpha^2 x + 1)\omega_0$$

$$F^{(6)} = xz + (\alpha^2 x^6 + \alpha^4 x^5 + \alpha^5 x^4 + \alpha^3 x^3 + \alpha^3 x^2 + \alpha^2 x + 1)\omega_0$$

\vdots

$$G^{(0)} = xz + (\alpha^3 x^6 + \alpha^4 x^5 + \alpha^4 x^4 + \alpha^3 x^3 + \alpha^2 x^2 + \alpha^3 x)\omega_0$$

$$F^{(0)} = (\alpha^3 x^6 + \alpha^5 x^5 + \alpha^4 x^4 + \alpha^3 x^3 + \alpha^2 x^2 + \alpha^3 x)\omega_0$$

and in the second phase, it computes $$m_3 = 1, m_2 = 1, m_1 = 0, m_0 = 1$$. Then from $$\mu = 1 + x^2 + x^3$$, the algorithm computes the corrected codeword $$(1, 1, 1, 0, 1, 0, 0, 0)$$ to output.

7. Final remarks

We presented a fast unique decoding algorithm for differential AG codes. The principle of the algorithm is the same with the companion algorithm for evaluation AG codes in [10], and the description of the algorithm is almost identical with that of the other. In this paper, we focused on the differences and omitted most of the repetitive parts. The list decoding extension for evaluation AG codes done by [4] can be done for differential AG codes in the same way. The results of Section 4.3 were used as a basis for the subsequent work [9].

Acknowledgments

This work was partially done while the author was visiting Maria Bras-Amorós. He thanks her for hospitality and stimulating discussions. Finally, he appreciate the valuable suggestions and criticisms of anonymous reviewers.

References

[1] P. Beelen and T. Høholdt, The decoding of algebraic geometry codes, in Advances in Algebraic Geometry Codes, World Sci. Publ., 2008, 49–98.

[2] I. M. Duursma, Majority coset decoding, IEEE Trans. Inf. Theory, 39 (1993), 1067–1070.

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

[4] O. Geil, R. Matsumoto and D. Ruano, Decoding algorithms based on Gröbner bases for general one-point AG codes, in Proc. IEEE Int. Symp. Inf. Theory, 2012, 86–90.

[5] O. Geil, R. Matsumoto and D. Ruano, Feng-Rao decoding of primary codes, Finite Fields Appl., 23 (2013), 35–52.

[6] O. Geil, C. Munuera, D. Ruano and F. Torres, On the order bounds for one-point AG codes, Adv. Math. Commun., 5 (2011), 489–504.

[7] V. D. Goppa, Codes on algebraic curves, Sov. Math. Dokl., 24 (1981), 170–172.

[8] T. Høholdt, J. H. van Lint and R. Pellikaan, Algebraic geometry of codes, in Handbook of Coding Theory, North-Holland, 1998, 871–961.

[9] K. Lee, Bounds for generalized Hamming weights of general AG codes, Finite Fields Appl., 34 (2015), 265–279.

[10] K. Lee, M. Bras-Amorós and M. E. O’Sullivan, Unique decoding of general AG codes, IEEE Trans. Inf. Theory, 60 (2014), 2638–2653.

[11] S. Sakata, H. E. Jensen and T. Høholdt, Generalized Berlekamp-Massey decoding of algebraic-geometric codes up to half the Feng-Rao bound, IEEE Trans. Inf. Theory, 41 (1995), 1762–1768.

[12] H. Stichtenoth, Algebraic Function Fields and Codes, 2nd edition, Springer-Verlag, 2009.

Received for publication May 2014.

E-mail address: kwankyu@chosun.ac.kr