Progressive Algebraic Soft Decoding of Hermitian Codes

Jiwei Liang[†], Li Chen[‡§],
[†] School of System Science and Engineering, Sun Yat-sen University, Guangzhou, China
[‡] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China
[§] Guangdong Provincial Key Laboratory of Information Security Technology, Guangzhou, China
Email: liangjw59@mail2.sysu.edu.cn, chenli55@mail.sysu.edu.cn

2026 IEEE International Symposium on Information Theory (ISIT) | 979-8-3315-8004-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/ISIT62367.2026.11654087

***Abstract*—THIS PAPER IS ELIGIBLE FOR THE STUDENT PAPER AWARD. This paper proposes the progressive algebraic soft decoding (PASD) for Hermitian codes. The decoding is formulated as finding a Gröbner basis of the interpolation module that contains the interpolation polynomial. By analyzing the algebraic structure of the interpolation module basis, size of the submodule basis can be progressively enlarged. This yields a progressive interpolation that can adapt the decoding complexity to the channel conditions. Our analysis shows that the proposed PASD can reduce the average decoding complexity compared to the prototype algebraic soft decoding (ASD).**

***Index Terms*—Algebraic-geometric codes, algebraic soft decoding, basis reduction, Hermitian codes**

I. Introduction

Algebraic-geometric (AG) codes are linear block codes constructed based on algebraic curves [1]. AG codes comprise a large family, including Reed-Solomon (RS) codes, elliptic codes and Hermitian codes. Among them, RS codes can be considered as a special case of AG codes, as they are constructed from a straight line. However, length of an RS code cannot exceed size of the finite field in which it is defined, limiting its error-correction capability. Compared with RS codes, general AG codes have greater codeword lengths, inheriting stronger error-correction capabilities.

Similar to RS codes, AG codes can be decoded by the syndrome-based decoding algorithms [2, 3]. However, they cannot correct errors beyond half of the code's minimum Hamming distance. By formulating the decoding as a curve-fitting problem, the Guruswami-Sudan (GS) algorithm can correct errors beyond this limit [4]. The GS decoding consists of interpolation and root-finding, where the former dominates the complexity. One approach to realize interpolation is through Kötter's algorithm [5] which constructs the interpolation polynomial in an iterative manner. The interpolation polynomial can be seen as the minimum candidate of the Gröbner basis of an interpolation module. Consequently, it can be obtained through a module basis construction and reduction. The basis reduction (BR) for GS decoding of RS codes was first proposed in [6]. It can be generalized to decode other AG codes, such as Hermitian codes [7] and elliptic codes [8, 9]. Although GS decoding has a better error-correction capability, its interpolation remains computationally expensive. It can be facilitated by transforming the received word to another one that contains more zero symbols. This is called the re-encoding transform (ReT). It was first introduced for RS codes [10] and recently generalized to decode general AG codes [11].

The error-correction capability of GS decoding can be improved by utilizing soft information obtained from the channel output rather than hard-decision symbols. Based on formulating test-vectors and exploiting their similarity, the low-complexity Chase (LCC) decoding was developed for RS codes [12] and elliptic codes [13, 14], respectively. For Hermitian codes, LCC decoding was first introduced in [15] and later improved in [16]. The algebraic soft decoding (ASD) is another approach that enhances the decoding performance by transforming soft information into interpolation multiplicities. It was first proposed for RS codes [17], and later extended for decoding Hermitian codes through Kötter's interpolation [18] and the BR interpolation [19], respectively. Following these developments, the BR interpolation-based ASD of elliptic codes was proposed in [20]. Although ASD yields substantial decoding performance improvements, its complexity remains high. Recent research [20, 21] shows that the ReT can facilitate the ASD of elliptic codes and Hermitian codes. Progressive interpolation was first introduced for the ASD of RS codes from another perspective, aiming to adapt the decoding complexity to the channel conditions [22, 23]. In the paradigm of BR interpolation, it can be realized through progressively enlarging size of the submodule basis until the intended message is decoded [24]. However, progressive interpolation has not been considered for AG codes, due to the challenges in its mathematical handling over function fields of algebraic curves.

This paper proposes a progressive algebraic soft decoding (PASD) for Hermitian codes. By analyzing the algebraic structure of the module basis, the proposed PASD progressively constructs the interpolation submodule basis, thereby adapting the decoding computation to channel conditions and reducing the average complexity. This progressive interpolation can be further simplified by employing a reduced basis. Complexity analysis shows that the proposed PASD reduces the average decoding complexity compared to its non-progressive prototype, i.e., the algebraic soft decoding (ASD).

II. Background Knowledge

This section presents the background knowledge of Hermitian codes, the GS decoding and the ASD.

A. Hermitian Codes

Let $\mathbb{F}_q = \{\sigma_0, \sigma_1, \ldots, \sigma_{q-1}\}$ denote the finite field of size q, where q is a square as required for Hermitian codes. Let $\mathbb{F}_q[X, Y]$ denote the bivariate polynomial ring defined over $\mathbb{F}_q$. An affine Hermitian curve defined over $\mathbb{F}_q$ is given by

$$H_w = Y^w + Y - X^{w+1}, \tag{1}$$

where $w = \sqrt{q}$ and the curve has a genus $g = w(w-1)/2$. There are w^3 affine points $P_j = (x_j, y_j)$ satisfying $H_w(x_j, y_j) = 0$, where $0 \le j \le w^3 - 1$, along with a point at infinity P_∞. Let $\mathcal{P}$ denote the set of w^3 affine points. The coordinate ring of H_w is

$$\mathcal{R} = \mathbb{F}_q[X, Y] / < Y^w + Y - X^{w+1} > . \tag{2}$$

Let x and y denote the residue classes of X and Y, respectively. The pole basis $\mathcal{L}_w$ comprises a set of bivariate monomials $\phi_a(x, y) = x^{i_x} y^{i_y}$, where $i_x, i_y \in \mathbb{N}$. For a nonzero polynomial $S \in \mathcal{R}$, its order at a rational point P is denoted as $\nu_P(S)$. Let $\nu_{P_\infty}(\phi_a^{-1})$ denote the pole order at P_∞ and $\nu_{P_\infty}(\phi_a^{-1}) = \nu_{P_\infty}((x^{i_x} y^{i_y})^{-1}) = w \cdot i_x + (w+1) \cdot i_y$, where $0 \le i_x \le w$ and $i_y \ge 0$. Monomials of the pole basis exhibit an increasing pole order as $\mathcal{L}_w = \{\phi_a | \nu_{P_\infty}(\phi_a^{-1}) < \nu_{P_\infty}(\phi_{a+1}^{-1})\}$. Let n and k denote the length and dimension of the code, respectively. Let $\mu = k + g - 1$ and $\mathcal{L}(\mu P_\infty)$ is the Riemann-Roch space defined by μ and P_∞. For an (n, k) Hermitian code, given a message polynomial

$$f(x, y) = f_0\phi_0 + f_1\phi_1 + \cdots + f_{k-1}\phi_{k-1} \in \mathcal{L}(\mu P_\infty), \tag{3}$$

where $f_0, f_1, \ldots, f_{k-1} \in \mathbb{F}_q$ are the message symbols, the codeword $\underline{c} \in \mathbb{F}_q^n$ is generated by

$$\underline{c} = (c_0, c_1, \ldots, c_{n-1}) = (f(P_0), f(P_1), \ldots, f(P_{n-1})), \tag{4}$$

where $\{P_0, P_1, \ldots, P_{n-1}\} \subseteq \mathcal{P}$. For an integer $a \ge 0$, define $[a] = \{0, 1, \ldots, a-1\}$. Thus, the index set of $\underline{c}$ is $[n]$.

B. The GS Decoding

For GS decoding of Hermitian codes, the following Definition is needed.

Definition 1: Let $\mathcal{R}[z]$ denote the polynomial ring defined over $\mathcal{R}$. Monomials $\phi_a z^b \in \mathcal{R}[z]$ are ordered according to their $(1, w_z)$-weighted degrees as

$$\deg_{1,w_z} \phi_a z^b = \nu_{P_\infty}(\phi_a^{-1}) + w_z b, \tag{5}$$

where $w_z = \nu_{P_\infty}(\phi_{k-1}^{-1})$. The $(1, w_z)$-reverse lexicographic (revlex) order can be established as follows. Given two monomials $\phi_{a_1} z^{b_1}$ and $\phi_{a_2} z^{b_2}$, $\mathrm{ord}(\phi_{a_1} z^{b_1}) < \mathrm{ord}(\phi_{a_2} z^{b_2})$, if $\deg_{1,w_z} \phi_{a_1} z^{b_1} < \deg_{1,w_z} \phi_{a_2} z^{b_2}$, or $\deg_{1,w_z} \phi_{a_1} z^{b_1} = \deg_{1,w_z} \phi_{a_2} z^{b_2}$ and $b_1 < b_2$. Given a polynomial $Q(x, y, z) = \sum_{a,b \in \mathbb{N}} Q_{ab} \phi_a(x, y) z^b$, the $(1, w_z)$-weighted degree of Q is $\deg_{1,w_z} Q = \max\{\deg_{1,w_z} \phi_a z^b \mid Q_{ab} \ne 0\}$ and its leading order is $\mathrm{lod}(Q) = \max\{\mathrm{ord}(\phi_a z^b) \mid Q_{ab} \ne 0\}$.

In decoding an (n, k) Hermitian code, polynomials are organized under the $(1, w_z)$-revlex order. Given two polynomials Q_1 and Q_2, we claim $Q_1 < Q_2$, if $\mathrm{lod}(Q_1) < \mathrm{lod}(Q_2)$. Let $\underline{\omega} = (\omega_0, \omega_1, \ldots, \omega_{n-1}) \in \mathbb{F}_q^n$ denote the received word. Interpolation points can be formed as

$$\mathbf{P} = \{(P_0, \omega_0), (P_1, \omega_1), \ldots, (P_{n-1}, \omega_{n-1})\}. \tag{6}$$

Theorem 1 *[4, 7]:* Given a polynomial f in the form of (3) and polynomial $\mathcal{Q}$ which has a zero of multiplicity at least m over the n interpolation points, if $m(n - |\{j \mid f(P_j) \ne w_j, j \in [n]\}|) > \deg_{1,w_z} \mathcal{Q}$, $\mathcal{Q}(x, y, f) = 0$, or equivalently $(z - f) | \mathcal{Q}$.

Interpolation constructs the polynomial $\mathcal{Q}$. The z-roots of $\mathcal{Q}$ are determined as the estimated message polynomials.

C. The ASD and Its BR Interpolation

This subsection introduces the ASD, which offers a flexible multiplicity assignment for improved utilization of soft information. We begin with the reliability transform.

1) *Reliability Transform*: Assume that a codeword $\underline{c}$ is transmitted and $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$ is the channel output. A reliability matrix $\mathbf{\Pi} \in \mathbb{R}^{q \times n}$ with entries $\pi_{ij} = \Pr(r_j \mid c_j = \sigma_i)$ can be obtained. Let l denote the designed maximum decoding output list size (OLS). Parametrized by l, $\mathbf{\Pi}$ will be transformed into a multiplicity matrix $\mathbf{M}$ of the same size, whose entries $m_{ij} \in \mathbb{Z}_0^+$ specify the multiplicity assigned to the point (P_j, σ_i). Let i_j be the index such that $\sigma_{i_j} = c_j$. The codeword score of $\mathbf{M}$ is defined as

$$s_{\mathbf{M}}(\underline{c}) = \sum_{j=0}^{n-1} m_{i_j j}. \tag{7}$$

2) *Interpolation and Root-Finding*: Let $\mathrm{mult}_{(P_j, \sigma_i)}(Q)$ denote the multiplicity of polynomial Q at point (P_j, σ_i). Given $\mathbf{M}$, the interpolation module $\mathcal{I}_{\mathbf{M},l}$ can be defined as[1]

$$\mathcal{I}_{\mathbf{M},l} = \{Q \in \mathcal{R}[z] \mid \mathrm{mult}_{(P_j, \sigma_i)}(Q) \ge m_{ij} \text{ and } \deg_z Q \le l \text{ for } 0 \le i \le q-1, 0 \le j \le n-1\}.$$

Given $\mathbf{M}$, interpolation aims to construct the minimum polynomial $\mathcal{Q}$ in $\mathcal{I}_{\mathbf{M},l}$ with respect to the $(1, w_z)$-revlex order.

Theorem 2 *[19]:* Given a polynomial f in the form of (3) and the interpolation polynomial $\mathcal{Q} \in \mathcal{I}_{\mathbf{M},l}$, if $s_{\mathbf{M}}(\underline{c}) > \deg_{1,w_z} \mathcal{Q}$, $\mathcal{Q}(x, y, f) = 0$, or equivalently $(z - f) | \mathcal{Q}$.

Polynomial $\mathcal{Q}$ can be determined by BR interpolation. It first derives a series of intermediate multiplicity matrices from $\mathbf{M}$. Let $\eta = \max_j\{\sum_{i=0}^{q-1} m_{ij}\}$. These intermediate multiplicity matrices are denoted as $\mathbf{M}^{(u)}$, where $u = 0, 1, \ldots, \eta$. The first matrix is set as $\mathbf{M}^{(0)} = \mathbf{M}$. Let $i_j^{(u)} = \arg\max_i\{m_{ij}^{(u)}\}$. Matrices $\mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \ldots, \mathbf{M}^{(\eta)}$ are determined sequentially, with their entries $m_{ij}^{(1)}, m_{ij}^{(2)}, \ldots, m_{ij}^{(\eta)}$ updated by

$$m_{ij}^{(u+1)} = \begin{cases} m_{ij}^{(u)} - 1, & \text{if } i = i_j^{(u)} \text{ and } m_{ij}^{(u)} \ne 0, \\ m_{ij}^{(u)}, & \text{if } i \ne i_j^{(u)} \text{ or } m_{ij}^{(u)} = 0. \end{cases} \tag{8}$$

[1]The interpolation module $\mathcal{I}_{\mathbf{M},l}$ is a submodule of the free module $\mathcal{R}[z]_l$. In the sequel, the term submodule (of this interpolation module) refers specifically to those submodules of $\mathcal{I}_{\mathbf{M},l}$ that correspond to different intermediate multiplicity matrices $\mathbf{M}^{(u)}$.

***Definition II**:* Given $\mathcal{J} \subseteq [n]$, $\mathbb{A}(\mathcal{J}) = \{x_j \mid j \in \mathcal{J}\}$. For $\sigma \in \mathbb{A}(\mathcal{J})$, $\mathbb{B}_\sigma(\mathcal{J}) = \{y_j \mid (\sigma, y_j) \in \mathcal{P},\ j \in \mathcal{J}\}$.

Given an affine point index set $\mathcal{J} \subseteq [n]$, the following polynomial can be defined

$$L_{\mathcal{J},j}(x,y) = \prod_{\alpha \in \mathbb{A}(\mathcal{J}) \setminus \{x_j\}} \frac{x-\alpha}{x_j-\alpha} \prod_{\beta \in \mathbb{B}_{x_j}(\mathcal{J}) \setminus \{y_j\}} \frac{y-\beta}{y_j-\beta}. \quad (9)$$

It satisfies $L_{\mathcal{J},j}(P_j) = 1$ and $L_{\mathcal{J},j}(P_{j'}) = 0$, for all distinct $j, j' \in \mathcal{J}$. Let $\underline{\omega}^{(u)} = (\omega_0^{(u)}, \omega_1^{(u)}, \ldots, \omega_{n-1}^{(u)})$, where $\omega_j^{(u)} = \sigma_{i_j^{(u)}}$. We define the following polynomial

$$K_{\underline{\omega}^{(u)}}(x,y) = \sum_{j \in [n]} \omega_j^{(u)} L_{[n],j}(x,y). \quad (10)$$

Note that $K_{\underline{\omega}^{(u)}}(P_j) = \omega_j^{(u)}$. Let $\mathfrak{m}_j^{(u)} = \max_i\{m_{ij}^{(u)}\}$, and $\mathcal{E}_{\mathbf{M}^{(u)}}$ be an ideal in $\mathcal{R}$ as $\mathcal{E}_{\mathbf{M}^{(u)}} = \{S \in \mathcal{R} \mid \nu_{P_j}(S) \geq \mathfrak{m}_j^{(u)},\ \text{for all } j\}$. To compute $\mathcal{E}_{\mathbf{M}^{(u)}}$, P_j is reassigned to $P_{a,b}$ where $a = \lfloor j/w \rfloor$ and $b = j \bmod w$. Subsequently, let $v_{a,b}^{(u)} = \mathfrak{m}_j^{(u)}$. For each a, index b is arranged such that $v_{a,b}^{(u)}$ are arranged in decreasing order as

$$v_{a,0}^{(u)} \geq v_{a,1}^{(u)} \geq \ldots \geq v_{a,w-1}^{(u)}. \quad (11)$$

Given $B_{b,c}^{(u)} \in \mathbb{F}_q[x]$ satisfying $\nu_{P_{a,b}}(y - B_{b,c}^{(u)}) \geq v_{a,b}^{(u)} - v_{a,c}^{(u)}$, we further define the following polynomial as the generator polynomial of $\mathcal{E}_{\mathbf{M}^{(u)}}$

$$T_{u,c}(x,y) = \prod_{a=0}^{q-1} (x-\sigma_a)^{v_{a,c}^{(u)}} \prod_{0 \leq b \leq c-1} (y - B_{b,c}^{(u)}), \quad (12)$$

where $0 \leq u \leq \eta$ and $0 \leq c \leq w-1$. The basis of $\mathcal{I}_{\mathbf{M},l}$ can be generated as follows.

***Theorem 3** [19]: $\mathcal{I}_{\mathbf{M},l}$ can be generated by*

$$\mathcal{M} = \{M_{u,c} \mid M_{u,c} = T_{u,c} \prod_{r=0}^{u-1} (z - K_{\underline{\omega}^{(r)}})\}. \quad (13)$$

The following Lemma shows that module $\mathcal{I}_{\mathbf{M}^{(u-1)},l}$ can be constructed by $\mathcal{I}_{\mathbf{M}^{(u)},l}$ together with the ideal $\mathcal{E}_{\mathbf{M}^{(u-1)}}$.

***Lemma 4** [19]:* Given two intermediate multiplicity matrices $\mathbf{M}^{(u-1)}$ and $\mathbf{M}^{(u)}$,

$$\mathcal{I}_{\mathbf{M}^{(u-1)},l} = \mathcal{I}_{\mathbf{M}^{(u)},l}(z - K_{\underline{\omega}^{(u-1)}}) + \mathcal{E}_{\mathbf{M}^{(u-1)}}. \quad (14)$$

Consequently, the submodule basis can also be constructed progressively. After the basis construction, the Mulders-Storjohann (MS) algorithm [25] can be applied to reduce $\mathcal{M}$ into a Gröbner basis, from which the minimum polynomial is chosen as the interpolation polynomial [2]. Then, the estimated message polynomial f_{est} is obtained by finding z-roots of $\mathcal{Q}$. This can be realized by the recursive coefficient search (RCS) algorithm [26, 27]. If there are multiple z-roots, the one whose corresponding codeword yields the minimum Hamming distance to $\underline{\omega}$ will be chosen as the decoding output.

[2]Although there are other algorithms that have better asymptotic complexity, the MS algorithm is more efficient for handling codes of practical length.

III. Progressive BR Interpolation

This section presents the progressive BR interpolation. It is substantiated by the progressive submodule basis construction and its subsequent reduction.

A. Progressive Submodule Basis Construction

To realize the progressive submodule basis construction, the following Definition is needed.

***Definition III**:* For integers a and b, the function $\Delta_{a,b}$ is defined as

$$\Delta_{a,b} = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

Let $\mathbf{M}$ and $\mathbf{M}'$ denote two multiplicity matrices where their entries m_{ij} and m'_{ij} satisfy

$$m'_{ij} = \begin{cases} m_{ij} - 1, & \text{if } i = i_j \text{ and } m_{ij} \neq 0, \\ m_{ij}, & \text{if } i \neq i_j \text{ or } m_{ij} = 0, \end{cases} \quad (16)$$

where $i_j = \arg\max_i\{m_{ij}\}$. Recall that $\mathcal{E}_{\mathbf{M}}$ is an ideal in $\mathcal{R}$, i.e. $\mathcal{E}_{\mathbf{M}} = \{S \in \mathcal{R} \mid \nu_{P_j}(S) \geq \mathfrak{m}_j\}$. Let $\mathcal{I}_{\mathbf{M}',l}$ denote the interpolation module with respect to $\mathbf{M}'$. Note that $\mathcal{I}_{\mathbf{M}',l}$ is a submodule of $\mathcal{I}_{\mathbf{M},l}$. The following Lemma shows how to construct the basis of $\mathcal{I}_{\mathbf{M},l}$ using the basis of $\mathcal{I}_{\mathbf{M}',l}$ and $\mathcal{E}_{\mathbf{M}}$.

***Lemma 5**:* The interpolation module $\mathcal{I}_{\mathbf{M},l}$ is generated by

$$\begin{aligned} \mathcal{M} = \{&M_{u,c} = \Delta_{u,0} T_{u,c} \\ &+ (1 - \Delta_{u,0}) M'_{u-1,c} \cdot (z - K_{\underline{\omega}^{(0)}})\}, \end{aligned} \quad (17)$$

where $M'_{u,c}$ is the submodule basis polynomial of $\mathcal{I}_{\mathbf{M}',l}$.

Proof: Based on (13), if $u = 0$, $M_{0,c} = T_{0,c}$. If $u \neq 0$, based on (16), we have

$$\mathbf{M}'^{(u-1)} = \mathbf{M}^{(u)}. \quad (18)$$

Let $i'^{(u)}_j = \arg\max_i\{m'^{(u)}_{ij}\}$ and $\mathfrak{m}'^{(u)}_j = \max_i\{m'^{(u)}_{ij}\}$. Further let $\mathcal{E}_{\mathbf{M}'}$ denote an ideal in $\mathcal{R}$, i.e. $\mathcal{E}_{\mathbf{M}'^{(u)}} = \{S \in \mathcal{R} \mid \nu_{P_j}(S) \geq \mathfrak{m}'^{(u)}_j\}$. Based on (18),

$$\mathcal{E}_{\mathbf{M}'^{(u-1)}} = \mathcal{E}_{\mathbf{M}^{(u)}}. \quad (19)$$

Their generators defined in (12) satisfy

$$T'_{u-1,c} = T_{u,c}. \quad (20)$$

Let $\omega'^{(u)}_j = \sigma_{i'^{(u)}_j}$ and $\underline{\omega}'^{(u)} = (\omega'^{(u)}_0, \omega'^{(u)}_1, \ldots, \omega'^{(u)}_{n-1})$. The following polynomial can be further defined as

$$K_{\underline{\omega}'^{(u)}}(x,y) = \sum_{j \in [n]} \omega'^{(u)}_j L_{[n],j}(x,y). \quad (21)$$

Since $\underline{\omega}'^{(u-1)} = \underline{\omega}^{(u)}$,

$$K_{\underline{\omega}'^{(u-1)}} = K_{\underline{\omega}^{(u)}}. \quad (22)$$

Therefore,

$$\begin{aligned} M'_{u-1,c} &= T'_{u-1,c}\prod_{r=0}^{u-2}(z-K_{\underline{\omega}^{(r)}}) \\ &= T_{u,c}\prod_{r=1}^{u-1}(z-K_{\underline{\omega}^{(r)}}). \end{aligned} \tag{23}$$

Based on (23), if $u \neq 0$,

$$\begin{aligned} M_{u,c} &= T_{u,c}\prod_{r=0}^{u-1}(z-K_{\underline{\omega}^{(r)}}) \\ &= M'_{u-1,c}\cdot(z-K_{\underline{\omega}^{(0)}}). \end{aligned} \tag{24}$$

□

Lemma 5 can also be generalized to the submodule basis construction of $\mathcal{M}^{(u-1)}$. This generalization is given in the following Corollary.

Corollary 6*:* For $1 \leq u \leq \eta-1$, $0 \leq u' \leq \eta-u+1$ and $0 \leq c \leq w-1$, the submodule basis $\mathcal{M}^{(u-1)}$ with respect to $\mathbf{M}^{(u-1)}$ can be generated by

$$\begin{aligned} \mathcal{M}^{(u-1)} = \{&M^{(u-1)}_{u',c} = \Delta_{u',0}T_{u-1,c} \\ &+ (1-\Delta_{u',0})M^{(u)}_{u'-1,c}\cdot(z-K_{\underline{\omega}^{(u-1)}})\}. \end{aligned} \tag{25}$$

Lemma 5 and *Corollary 6* imply that the submodule bases can be constructed progressively. Thus, we do not need to perform decoding directly for $\mathbf{M}$. A series of intermediate multiplicity matrices $\mathbf{M}^{(u)}$ are generated based on (8), where $u = 0, 1, \ldots, \eta$. Since $\mathbf{M}^{(\eta)}$ is an all-zero matrix, the submodule basis with respect to $\mathbf{M}^{(\eta-1)}$, denoted by $\mathcal{M}^{(\eta-1)}$, can first be constructed as in (13)[3]. The MS algorithm is applied to reduce this basis into a Gröbner basis, where $\mathcal{Q}$ can be obtained. The estimated message polynomial is determined by finding its z-roots. If this step succeeds, the decoding terminates and delivers its output[4]. If the decoding fails, $\mathcal{M}^{(\eta-1)}$ is applied to construct $\mathcal{M}^{(\eta-2)}$ using (25). The subsequent basis reduction and root-finding will be performed. This progressive iteration continues until either decoding succeeds for some $\mathbf{M}^{(u)}$, or decoding fails for $\mathbf{M}^{(0)}$, (i.e., when the maximum number of progressive iterations is reached). When channel conditions improve, the decoding can succeed with fewer interpolation constraints and terminate early.

B. Progressive Construction Using Reduced Submodule Basis

In the progressive interpolation described above, generating $\mathcal{M}^{(u-1)}$ requires storing $\mathcal{M}^{(u)}$, which incurs additional memory cost. Let $\Theta^{(u)} = \{\theta^{(u)}_{u',c}\}$ denote the reduced Gröbner basis derived from $\mathcal{M}^{(u)}$. The following Lemma further reveals that $\mathcal{M}^{(u-1)}$ can also be generated based on $\Theta^{(u)}$.

[3]The PASD is presented as starting from $\mathcal{M}^{(\eta-1)}$ to maintain generality across all multiplicity assignment methods. For some methods (e.g., Algorithm A of [17]), decoding may not succeed immediately at this stage due to weak constraints. Similarly, the maximum number of progressive iterations is determined by the specific multiplicity assignment method and may exceed l.

[4]Decoding succeeds if the codeword yielded by the estimated message polynomial satisfies the maximum likelihood (ML) criterion [28, 29].

Lemma 7*:* The submodule basis $\mathcal{M}^{(u-1)}$ with respect to $\mathbf{M}^{(u-1)}$ can be generated by

$$\begin{aligned} \mathcal{M}^{(u-1)} = \{&M^{(u-1)}_{u',c} = \Delta_{u',0}T_{u-1,c} \\ &+ (1-\Delta_{u',0})\theta^{(u)}_{u'-1,c}\cdot(z-K_{\underline{\omega}^{(u-1)}})\}, \end{aligned} \tag{26}$$

where $1 \leq u \leq \eta-1$, $0 \leq u' \leq \eta-u+1$ and $0 \leq c \leq w-1$.

Proof: Since all basis polynomials $\theta^{(u)}_{u'-1,c}$ in $\Theta^{(u)}$ satisfy $\mathrm{mult}_{(P_j,\sigma_i)}(\theta^{(u)}_{u'-1,c}) \geq m^{(u)}_{ij}$,

$$\mathrm{mult}_{(P_j,\sigma_i)}(z-K_{\underline{\omega}^{(u-1)}}) \geq m^{(u-1)}_{ij} - m^{(u)}_{ij} \tag{27}$$

and

$$\mathrm{mult}_{(P_j,\sigma_i)}(\theta^{(u)}_{u'-1,c}(z-K_{\underline{\omega}^{(u-1)}})) \geq m^{(u-1)}_{ij}. \tag{28}$$

Therefore, $\mathcal{M}^{(u-1)}$ belongs to the module defined by $\mathbf{M}^{(u-1)}$.

We continue to prove that $\mathcal{M}^{(u-1)}$ is the submodule basis with respect to $\mathbf{M}^{(u-1)}$. For any polynomial Q in the module defined by $\mathbf{M}^{(u-1)}$,

$$Q = \sum_{s=0}^{\eta-u+1} Q_{[s]}z^s, \tag{29}$$

where $Q_{[s]} \in \mathcal{E}_{\mathbf{M}^{(s)}}$. There exist $\zeta^{(s)}_0, \zeta^{(s)}_1, \ldots, \zeta^{(s)}_{w-1} \in \mathbb{F}_q[x]$, such that

$$Q_{[s]} = \sum_{c=0}^{w-1}\zeta^{(s)}_c T_{s,c}. \tag{30}$$

It enables

$$Q^{(\eta-u)} = Q - \sum_{c=0}^{w-1}\zeta^{(\eta-u+1)}_c M^{(u-1)}_{\eta-u+1,c}, \tag{31}$$

where $\deg_z Q^{(\eta-u)} \leq \eta-u$. Again, there exist $\zeta^{(\eta-u)}_0, \zeta^{(\eta-u)}_1, \ldots, \zeta^{(\eta-u)}_{w-1} \in \mathbb{F}_q[x]$ such that

$$Q^{(\eta-u-1)} = Q^{(\eta-u)} - \sum_{c=0}^{w-1}\zeta^{(\eta-u)}_c M^{(u-1)}_{\eta-u,c}, \tag{32}$$

where $\deg_z Q^{(\eta-u-1)} \leq \eta-u-1$. This deduction continues, leading to the existence of $\zeta^{(1)}_0, \zeta^{(1)}_1, \ldots, \zeta^{(1)}_{w-1} \in \mathbb{F}_q[x]$ such that

$$Q^{(0)} = Q^{(1)} - \sum_{c=0}^{w-1}\zeta^{(1)}_c M^{(u-1)}_{1,c}. \tag{33}$$

Therefore, polynomial $Q^{(0)}$ can be written as $Q^{(0)} = \sum_{c=0}^{w-1}\zeta^{(0)}_c M^{(u-1)}_{0,c}$, where $\zeta^{(0)}_0, \zeta^{(0)}_1, \ldots, \zeta^{(0)}_{w-1} \in \mathbb{F}_q[x]$. Any polynomial Q in the module defined by $\mathbf{M}^{(u-1)}$ can be expressed as an $\mathbb{F}_q[x]$-linear combination of $M^{(u-1)}_{u',c}$. □

The basis reduction complexity is determined by the weighted degree of the maximum polynomial in the submodule basis [30]. Since $\Theta^{(u)}$ is reduced from $\mathcal{M}^{(u)}$, its weighted degree is not greater than that of $\mathcal{M}^{(u)}$. Therefore, for the basis reduction of $\mathcal{M}^{(u-1)}$, performing it as in (26) has lower complexity than performing it as in (25).

IV. Decoding Complexity Analysis

This section analyzes the complexity of the proposed PASD approach. Since the finite field multiplications dominate the arithmetic operations, the complexity is measured as the number of finite field multiplications in a decoding event.

A. Theoretical Analysis

Since PASD can succeed with fewer interpolation constraints and terminate early, we analyze its complexity for a given number of progressive iterations. Let ξ denote the number of progressive iterations in a decoding event. At the ξ-th progressive iteration, complexity of the progressive BR interpolation can be characterized as follows.

Lemma 8: The complexities of the basis construction and reduction are $O(\xi^6 w^2 n^2)$ and $O(\xi^5 w^2 n^3)$, respectively.

Proof: The basis construction involves computing $T_{u-1,c}$, $K_{\underline{\omega}^{(u-1)}}$, and $M^{(u-1)}_{u',c}$. Since $\deg_x \prod_{a=0}^{q-1}(x-\sigma_a)^{v^{(u-1)}_{a,c}} \leq \xi w^2$, $\deg_x \prod_{0\leq b\leq c-1}(y - B^{(u)}_{b,c}) \leq \xi w^3$ and $\deg_y \prod_{0\leq b\leq c-1}(y - B^{(u)}_{b,c}) \leq w-1$, computing $T_{u-1,c}$ exhibits a complexity of $O(\xi^2 n^2)$. As $L_{[n],j}$ can be pre-computed, computing $K_{\underline{\omega}^{(u-1)}}$ has a complexity of $O(n^2)$. Because $\deg_x \theta^{(u)}_{u',c} \leq \xi(n/w - 1)$, $\deg_y \theta^{(u)}_{u',c} \leq \xi(w-1)$ and $\deg_z \theta^{(u)}_{u',c} \leq \xi - 1$, $M^{(u-1)}_{u',c}$ can be determined with a complexity of $O(\xi^5 w n^2)$. Therefore, the complexity of basis construction is $O(\xi^6 w^2 n^2)$.

With the MS algorithm, the basis reduction complexity is mainly determined by the maximum weighted degree of the module basis polynomial [30]. The basis reduction complexity can be bounded by $O(\xi^3 w^3 \gamma^2)$, where γ is the maximum weighted degree of the module basis polynomial. Since $\gamma = O(\xi w n)$, the basis reduction complexity is $O(\xi^5 w^2 n^3)$. □

The prototype ASD constructs the interpolation module with respect to $\mathbf{M}$. Therefore, complexity of ASD is obtained by replacing the factor ξ in the above complexity formulation with η. Since the basis reduction complexity dominates the overall complexity and $\xi \leq \eta$, the proposed PASD reduces the complexity by a factor of $1-(\xi/\eta)^5$. Note that at the ξ-th progressive iteration, complexity of root-finding is $O(\xi k n)$. Therefore, although the PASD may perform multiple basis reduction and root-finding, this extra cost can be compensated on average by the reduction in the number of progressive iterations required under better channel conditions.

B. Numerical Results

Our simulations were conducted over the additive white Gaussian noise (AWGN) channel using binary phase shift keying (BPSK) modulation. For the PASD, the multiplicity matrix $\mathbf{M}$ is generated using Algorithm A of [17] that is parametrized by a predefined decoding OLS l [19]. The PASD is compared with the prototype ASD of [19], the ReT-based ASD (marked as ASD (ReT)) and its improved variant (marked as ASD (I-ReT)) of [21]. Note that under improved channel condition, the ASD (I-ReT) can employ more re-encoding points, resulting in a greater complexity reduction. The four decoding algorithms employ BR interpolation.

TABLE I
Comparison of the ASD and PASD in decoding the (64, 49) Hermitian code with $l = 4$

SNR (dB)	Avg. Dec. Complexity		Avg. Iter. (PASD)
	ASD	PASD	
4	5.44×10^6	5.45×10^6	5.15
5	3.58×10^6	1.52×10^6	3.45
6	3.12×10^6	3.54×10^5	2.40
7	1.85×10^6	1.25×10^5	2.02
8	1.03×10^6	9.59×10^4	2.00

TABLE II
Average complexity in decoding the (64, 49) Hermitian code with $l = 4$

SNR (dB)	ASD (ReT)	ASD (I-ReT)	PASD
4	$\mathbf{2.63 \times 10^6}$	2.68×10^6	5.45×10^6
5	1.57×10^6	1.67×10^6	$\mathbf{1.52 \times 10^6}$
6	1.18×10^6	1.07×10^6	$\mathbf{3.54 \times 10^5}$
7	7.82×10^5	4.52×10^5	$\mathbf{1.25 \times 10^5}$
8	4.03×10^5	1.63×10^5	$\mathbf{9.59 \times 10^4}$

Table I compares the decoding complexity of the prototype ASD and the proposed PASD for the (64, 49) Hermitian code with $l = 4$. At low signal-to-noise ratio (SNR), the two algorithms exhibit similar complexity. As the SNR increases, complexity of both algorithms decreases. This reduction is more pronounced for PASD as it requires fewer progressive iterations. This is also evidenced by the numerical results that show the average number of progressive iterations in different SNRs. Note that since the multiplicity assignment follows Algorithm A of [17], the PASD rarely succeeds at the first progressive iteration. At high SNR, the decoding tends to succeed at the second progressive iteration.

Table II compares the complexity of the proposed PASD with that of the two ReT-facilitated ASD. In this table, the best-performing algorithm for each SNR is highlighted in grey. At low SNR, both ReT-facilitated ASD yield a lower complexity than the PASD, since the PASD typically requires the maximum number of progressive iterations. However, as the SNR increases, the complexity advantage of the PASD becomes evident. This is because ReT reduces complexity at the level of n by extracting common factors. In contrast, based on *Lemma 8*, PASD reduces the number of progressive iterations ξ, which appears as a higher-order term in the complexity characterization. This reduction has a more pronounced impact on the complexity. Although ASD (I-ReT) also yields a significant complexity reduction at high SNR by employing more re-encoding points, PASD achieves a steeper decline starting from a lower SNR. Note that our recent research has shown that the ReT can be incorporated with the proposed PASD to further reduce the decoding complexity. Due to space limits, this combination is not covered in this manuscript.

REFERENCES

[1] V. Goppa, "Codes associated with divisors," *Problemy Peredachi Informatsii*, vol. 13, no. 1, pp. 33–39, 1977.

[2] J. Massey, "Shift register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. 15, no. 1, pp. 122–127, Jan. 1969.

[3] S. Sakata, J. Justesen, Y. Madelung, H. Jensen, and T. Høholdt, "Fast decoding of algebraic-geometric codes up to the designed minimum distance," *IEEE Trans. Inform. Theory*, vol. 41, no. 6, pp. 1672–1677, Nov. 1995.

[4] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. 45, no. 6, pp. 1757–1767, Sept. 1999.

[5] R. Kötter, *On Algebraic Decoding of Algebraic-geometric and Cyclic Codes*. Ph. D Thesis, Univ. Linköping, Linköping, Sweden, 1996.

[6] K. Lee and M. O'Sullivan, "List decoding of Reed-Solomon codes from a Gröbner basis perspective," *J. Symb. Comput.*, vol. 43, no. 9, pp. 645-658, Sept. 2008.

[7] K. Lee and M. O'Sullivan, "List decoding of Hermitian codes using Gröbner bases," *J. Symb. Comput.*, vol. 44, no. 12, pp. 1662-1675, Nov. 2008.

[8] P. Beelen and K. Brander, "Efficient list decoding of a class of algebraic-geometry codes," *Advances in Mathematics of Communications*, vol. 4, no. 4, pp. 485-518, Nov. 2010.

[9] Y. Wan, L. Chen, and F. Zhang, "Guruswami-Sudan decoding of elliptic codes through module basis reduction," *IEEE Trans. Inform. Theory*, vol. 67, no. 11, pp. 7197–7209, Nov. 2021.

[10] R. Kötter, J. Ma, and A. Vardy, "The re-encoding transformation in algebraic list-decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 633–647, Feb. 2011.

[11] Y. Wan, J. Xing, Y. Huang, T. Wu, B. Bai and G. Zhang, "The re-encoding transform in algebraic list decoding of algebraic geometric codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Taipei, Taiwan, Jun. 2023, pp. 19-24.

[12] J. Bellorado and A. Kavcic, "Low-complexity soft-decoding algorithms for Reed-Solomon codes - part I: an algebraic soft-in hard-out Chase decoder," *IEEE Trans. Inform. Theory*, vol. 56, no. 3, pp. 945–959, Mar. 2010.

[13] Y. Wan, J. Liang, L. Chen, and F. Zhang, "Low-complexity Chase decoding of elliptic codes," *IEEE Trans. Commun.*, *IEEE Trans. Commun.*, vol. 73, no. 10, pp. 8574-8586, Oct. 2025.

[14] J. Zhao and L. Chen, "Algebraic Chase decoding of elliptic codes with generalized re-encoding transform and improved root-finding", *IEEE Trans. Inform. Theory*, vol. 71, no. 7, pp. 5089-5108, Jul. 2025.

[15] S. Wu, L. Chen, and M. Johnson, "Interpolation-based low-complexity Chase decoding algorithms for Hermitian codes," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1376–1385, Apr. 2018.

[16] J. Liang, J. Zhao, and L. Chen, "Low-complexity Chase decoding of Hermitian codes with improved interpolation and root-finding," *IEEE Trans. Commun.*, vol. 73, no. 8, pp. 5509–5522, Aug. 2025.

[17] R. Kötter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.

[18] L. Chen, R. Carrasco, and M. Johnson, "Soft-decision list decoding of Hermitian codes," *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2169-2176, Aug. 2009.

[19] K. Lee and M. O'Sullivan., "Algebraic soft-decision decoding of Hermitian codes," *IEEE Trans. Inform. Theory*, vol. 56, no. 6, pp. 2587-2600, May 2010.

[20] Y. Wan, L. Chen and F. Zhang, "Algebraic soft decoding of elliptic codes," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1522-1534, Mar. 2022.

[21] J. Liang, and L. Chen, "Algebraic soft decoding of Hermitian codes with re-encoding transform," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Ann Arbor, USA, Jun. 2025.

[22] S. Tang, L. Chen and X. Ma, "Progressive list-enlarged algebraic soft decoding of Reed-Solomon codes", *IEEE Commun. Lett.*, vol. 16, no. 6, pp. 901-904, Jun. 2012.

[23] L. Chen, S. Tang and X. Ma, "Progressive algebraic soft-decision decoding of Reed-Solomon codes", *IEEE Trans. Commun.*, vol. 61, no. 2, pp. 433-442, Feb. 2013.

[24] Xing, L. Chen, M. Bossert, "Progressive algebraic soft-decision decoding of Reed-Solomon codes using module minimization", *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7379-7391, Nov. 2019.

[25] T. Mulders and A. Storjohann, "On lattice reduction for polynomial matrices," *J. Symb. Comput.*, vol. 35, no. 4, pp. 377–401, Apr. 2003.

[26] R. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Trans. Inform. Theory*, vol. 46, no. 1, pp. 246-257, Jan. 2000.

[27] X. Wu and P. Siegel, "Efficient root-finding algorithm with application to list decoding of algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 6, pp. 2579-2587, Sept. 2001.

[28] T. Kaneko, T. Nishijima, H. Inazumi, and S. Hirasawa, "An efficient maximum-likelihood-decoding algorithm for linear block codes with algebraic decoder," *IEEE Trans. Inform. Theory*, vol. 40, no. 2, pp. 320–327, Mar. 1994.

[29] X. Ma, and S. Tang, "Correction to "an efficient maximum-likelihood-decoding algorithm for linear block codes with algebraic decoder" [Mar 94 320-327]," *IEEE Trans. Inform. Theory*, vol. 58, no. 6, pp. 4073, Jun. 2012.

[30] J. Nielsen and P. Beelen, "Sub-quadratic decoding of one-point Hermitian codes," *IEEE Trans. Inform. Theory*, vol. 61, no. 6, pp. 3225-3240, Jun. 2015.