A generator of algebraic quantum error correction codes

Ichio Kikuchi[1], Akihito Kikuchi[2*]

[1]Internationales Forschungszentrum für Quantentechnik

[2]International Research Center for Quantum Technology, Tokyo

March 9, 2026

Abstract

In this article, we report the development of a computer program to build quantum error correction codes (QECC). It enables us to generate a QECC from an algebraic geometry code (AGC). We demonstrate how to incarnate the abstract idea (by which AG codes are extended to QEC codes) into an executable program, which systematically yields QECC over a range of adjustable parameters.

1 Introduction

This paper explains how to construct quantum error correction codes (QEC codes) systematically, using classical algebraic geometry codes (AG codes) as building blocks. The contents are composed of the following parts: first, we review the preliminary things for classical AG code; second, we learn how to extend an AG code to a QEC code; third, we study the coding of a program, utilizing existing resources (which are available as the functions implemented in the libraries of symbolic computation packages). The computer program used in this study is available at the authors' GitHub.

2 Preliminaries

2.1 Constructing Algebraic Geometry Codes Using the Riemann–Roch Theorem

In this section, we provide a brief overview of algebraic geometry (AG) codes. Concerning the details of the symbolic computation to compute the classical AG codes, the reader could consult several references, such as [BCL23, Gre17, CF02].

*akihito_kikuchi@gakushikai.jp (The corresponding author; a visiting researcher in IFQT)

Algebraic Geometry (AG) codes generalize Reed–Solomon codes by using rational points on algebraic curves over finite fields. The construction relies on the **Riemann–Roch theorem**, which provides the dimension of spaces of rational functions with prescribed zeros and poles.

Let C be a smooth projective curve over a finite field $\mathbb{F}_q$ with genus g. Let $P_1, \ldots, P_n$ be distinct rational points on C, and let D be a divisor on C disjoint from these points. The AG code is obtained by evaluating functions from the Riemann–Roch space $L(D)$ at these points.

2.1.1 Definitions

- **Divisor**: A formal sum $D = \sum m_i Q_i$ with $m_i \in \mathbb{Z}$ and Q_i points on C.
- **Riemann–Roch space**:

$$L(D) = \{f \in \mathbb{F}_q(C) : \mathrm{Div}(f) + D \geq 0\} \cup \{0\}.$$

- **Riemann–Roch theorem**:

$$\dim L(D) = \deg(D) - g + 1 \quad \text{for } \deg(D) > 2g - 2.$$

2.1.2 Steps to Construct an AG Code

1. Choose a projective curve C over $\mathbb{F}_q$ with many rational points and select distinct rational points $P_1, \ldots, P_n$.
2. Define a divisor D such that $\deg(D) < n$ and $\deg(D) > 2g - 2$. This ensures $L(D)$ has dimension $k = \deg(D) - g + 1$.
3. Compute $L(D)$ using the Brill–Noether method:
   (a) Compute the adjoint divisor A of C.
   (b) Find a homogeneous polynomial H such that $\mathrm{Div}(H) \geq D + A$.
   (c) Compute $\mathrm{Div}(H) - D$.
   (d) Solve a linear system to find homogeneous polynomials $G_1, \ldots, G_\ell$ of degree $\deg(H)$ satisfying $\mathrm{Div}(G_i) \geq \mathrm{Div}(H) - D$.
   (e) The basis of $L(D)$ is $\{G_i/H\}$.
4. Define the evaluation map:

$$C_L(D, P_1, \ldots, P_n) = \{(f(P_1), \ldots, f(P_n)) : f \in L(D)\} \subseteq \mathbb{F}_q^n.$$

2.1.3 Parameters of the Code

- Length: n (number of evaluation points).
- Dimension: $k = \dim L(D) \approx \deg(D) - g + 1$.
- Minimum Distance: $d \geq n - \deg(D)$.

2.1.4 Example: Hermitian Curve

Consider the Hermitian curve over $\mathbb{F}_{q^2}$:

$$C : y^q + y = x^{q+1}.$$

It has genus $g = \frac{q(q-1)}{2}$ and $q^3 + 1$ rational points. Choosing $D = mP_\infty$ and n rational points gives an AG code with excellent parameters.

2.1.5 Summary Algorithm (Pseudocode)

```
Input: Curve C over F_q, rational points {P_1,...,P_n}, divisor D.
Output: AG code C_L(D,P_1,...,P_n).

1. Compute adjoint divisor A of C.
2. Find H such that Div(H) >= D + A.
3. Compute Div(H) - D.
4. Solve linear system for G_i satisfying Div(G_i) >= Div(H) - D.
5. Basis of L(D): {G_i / H}.
6. Evaluate basis at points P_1,...,P_n.
```

2.2 Quantum error correction codes (QECC)

In this section, we review some preliminaries used for constructing quantum error-correcting codes [Gra20, CRSS97, AK00, KKKS05, DGG13, MTT16, GP17].

- A subspace $\mathcal{C}$ made of an orthonormal basis $\{|c_i\rangle\}$ in a Hilbert space $\mathcal{H}$ is a quantum error correction code for a quantum channel with error operators $\{E_k\}$ if and only if the following conditions are satisfied for all i, j, k, ℓ:

$$\langle c_i | E_k^\dagger E_\ell | c_j \rangle = 0 \text{ for } i \neq j \tag{1}$$

$$\langle c_i | E_k^\dagger E_\ell | c_i \rangle = \langle c_j | E_k^\dagger E_\ell | c_j \rangle = \alpha_{k\ell} \tag{2}$$

  Through the orthogonal projection $P_\mathcal{C} = \sum_i |c_i\rangle\langle c_i|$, those conditions are equivalent to

$$P_\mathcal{C} E_k^\dagger E_\ell P_\mathcal{C} = \alpha_{k\ell} P_\mathcal{C}. \tag{3}$$

- We use *qudit*, which is a Hilbert space spanned by an orthogonal basis made of $|0\rangle, |d-1\rangle$ with $d = p^m$, where the dimension d is a power of a prime number p. Then it is reasonable to label the basis states by the numbers in $\mathbb{F}_{p^n}$.

- We use the following operators, called local error basis:

$$X^a = \sum_{x \in \mathbb{F}_q} |x + a\rangle\langle x|, \tag{4}$$

$$Z^b = \sum_{y \in \mathbb{F}_q} \omega_p^{\mathrm{tr}(by)} |y\rangle\langle y|, \tag{5}$$

where $\omega_p = \exp(2\pi i/p)$. The notation tr is the trace used in the theory of the extension of algebraic fields. Those operators could represent the errors that occur in the quantum states that encode some information. Those errors are a generalization of bit-flip and phase errors in qubit systems. Those operators have the following property regarding the conjugation:

$$X^{-a} Z^b X^a = \left(\sum_{z \in \mathbb{F}_q} |z - a\rangle\langle z| \right) \left(\sum_{y \in \mathbb{F}_q} \omega_p^{\mathrm{tr}(by)} |y\rangle\langle y| \right) \left(\sum_{x \in \mathbb{F}_q} |x + a\rangle\langle x| \right) \tag{6}$$

$$= \sum_{x,y,z \in \mathbb{F}_q} \omega_p^{\mathrm{tr}(by)} |z - a\rangle\langle z|y\rangle\langle y|x + a\rangle\langle x| \tag{7}$$

$$= \sum_{x \in \mathbb{F}_q} \omega_p^{\mathrm{tr}(b(x+a))} |x\rangle\langle x| = \omega_p^{\mathrm{tr}(ab)} Z^b, \tag{8}$$

$$Z^b X^a = X^a \left(X^{-a} Z^b X^a \right) = \omega_p^{\mathrm{tr}(ab)} X^a Z^b, \tag{9}$$

and

$$X^a Z^b X^{a'} Z^{b'} = \omega_p^{\mathrm{tr}(a'b)} X^a X^{a'} Z^b Z^{b'} = \omega_p^{\mathrm{tr}(a'b)} X^{a'} X^a Z^{b'} Z^b \tag{10}$$

$$= \omega_p^{\mathrm{tr}(a'b - ab')} X^{a'} Z^{b'} X^a Z^b. \tag{11}$$

- We define the generalized n-qudit group composed of these operators. The elements of the group are written as

$$\mathcal{P}_n = \{\omega_p^\gamma X^{a_1} Z^{b_1} \otimes \ldots \otimes X^{a_n} Z^{b_n} \colon a_i, b_i \in \mathbb{F}_q, \gamma = 0, \ldots, p-1\}. \tag{12}$$

The order of this group is $|\mathcal{P}_n| = pq^{2n}$. Let $\mathcal{CP}_n$ be the center of P_n. The quotient group $\mathcal{P}_n/\mathcal{CP}_n$ is isomorphic to the vector space $\mathbb{F}_q^{2n}$ (in other words, an additive group) through a homomorphism

$$\phi \colon \omega_p^\gamma X^{a_1} Z^{b_1} \otimes \ldots \otimes X^{a_n} Z^{b_n} \mapsto (a_1, \ldots a_n | b_1, \ldots, b_n) = (\mathbf{a}|\mathbf{b}). \tag{13}$$

- Let $\mathcal{CP}_n$ be the center of $\mathcal{P}_n$. The quotient group is isomorphic to the vector space. As seen in the commutator relation, the exchange of operators always produces the phase factor $\mathrm{tr}(\mathbf{a}\cdot\mathbf{b})$

- The weight of an error operator E, which is acting on $\mathbb{C}_q^n = \mathbb{C}_q \otimes ... \otimes \mathbb{C}q$ is the number of subsystems on which E acts non-trivially. In other words, this definition corresponds to the Hamming weight of an error; it is the number of positions in the code where changes occur. By this definition, the weight of an operator is given by

$$\mathrm{wgt}\left(X^{\mathbf{a}}Z^{\mathbf{b}}\right) = |\{i|(a_i,b_i) \neq (0,0) \text{ for } i = 1,..,n\}|. \tag{14}$$

- A stabilizer quantum error-correcting code $C = ((n,K,d))_q$ is the eigenspace (with eigenvalue 1) of an Abelian subgroup S of the n-qudit Pauli group P_n: it does not contain a non-trivial multiple of identity. The dimension of the code is $K = \mathrm{dim}C = q^n/|S|$.

- The Euclidean dual code for an error correction code $C$is defined as

$$C^{\perp} = \{(x_0,x_1,...,x_{n-1}) \in \mathbb{F}_q^n | \sum_{i=0} x_i c_i = 0 \text{ for all } (c_0,c_1,...,c_n) \in \mathcal{C}\} \tag{15}$$

  where $\mathcal{C}$ stands for the codewords of C.

- For $\mathbb{F}_{q^2}$–linear codes, the inner product $*$ is replaced by a Hermitian form $x * y = \sum_{i=1}^n x_i y_i^q$. For these codes, we use the Hermitian dual defined as

$$C^h = \{(x_0,x_1,...,x_{n-1}) \in |\mathbb{F}_{q^2}^n| \sum_{i=0} x_i c_i^q = 0 \text{ for all } (c_0,c_1,...,c_n) \in \mathcal{C}\}. \tag{16}$$

  With this form, the codes are self-orthogonal.

- It requires a bit of discussion to see the validity of the use of the Hermitian dual [KKKS05].

  Let (β,β^q) denote the normal basis of $\mathbb{F}_{q^2}$ over $\mathbb{F}_q$. We define a trace alternating form of two vectors v and w in $\mathbb{F}_q^n$ by

$$\langle v|w\rangle_a = \mathrm{tr}_{q^2/q}\left(\frac{v\cdot w^q - v^q\cdot w}{\beta^{2q}-\beta^2}\right). \tag{17}$$

  We define a bijective map ϕ:

$$\phi((a|b)) = \beta a + \beta^q b, \tag{18}$$

  which transforms the element $(a|b)$ in $\mathbb{F}_q^{2n}$ into a vector in $\mathbb{F}_{q^2}$. Those definitions enable us to construct the symplectic form as follows:

$$\langle (a|b)\,|\,(a'|b')\rangle_s = \langle \phi((a|b))\,|\,\phi((a'|b'))\rangle_a = \mathrm{tr}_{q^2/q}(b\cdot a' - b'\cdot a). \tag{19}$$

Then we can use the simple inner product defined by $v \cdot w^q$ to check the orthogonality between two code vectors in the QECC, where q is set to a prime p.

In short, using the above ideas, we can construct a classical $\mathbb{F}_{p^2}$ code, where the code vectors are represented as $\phi((a|b))$. As ϕ is a bijective map, we can get the code vector $(a|b)$ of QECC.

Note that there is an alternative way. We can use the following trace alternating form

$$\langle v|w\rangle_{a'} = \mathrm{tr}_{q^2/q}\left(\frac{v \cdot w^q - v^q \cdot w}{\gamma - \gamma^q}\right). \tag{20}$$

with the polynomial basis γ of $\mathbb{F}_{q^2}/\mathbb{F}_q$ and $\phi((a|b)) = a + \gamma b$. With this definition, one can construct another sympletic form.

2.3 The conversion from classical codes to quantum ones

The method for extending a classical code to a quantum one is given in [DGG13]. It is an embedding of the matrix (which represents the classical code) in a larger one.

We use the common notation as in the previous sections. C and $C^{\perp_h}$ are the codewords and their Hermitian dual.

Let us define

$$e = \dim(C^{\perp_h}) - \dim(C \cap C^{\perp_h}) = \dim(C + C^{\perp_h}) - \dim(C), \tag{21}$$

and

$$s = \dim(C \cap C^{\perp_h}). \tag{22}$$

Let G be the matrix:

$$G = \begin{pmatrix} M_{s\times n} & 0_{s\times e} \\ A_{(n-e-2s)\times n} & 0_{(n-e-2s)\times e} \\ B_{e\times n} & \beta I_{e\times e} \end{pmatrix}. \tag{23}$$

The subscripts indicate the sizes of the sub-matrices, and 0 and I denote the zero matrix and the identity matrix, respectively. β is a number in $\mathbb{F}_{q^2}$ such that $\beta\bar{\beta} = \beta^{q+1} = -1$.

The interpretation of this construction is as follows.

- For a matrix P, let $r(P)$ denote the set of rows of P. The matrix G is constructed so that $r(M)$ is a basis for $C \cap C^{\perp_h}$, $r(M) \cup r(A)$ is a basis for C, $r(M) \cup r(B)$ is a basis for $C^{\perp_h}$, and $r(B)$ is an orthonormal set. Note that $r(M) \cup r(A) \cup r(B)$ is a basis for $C + C^{\perp_h}$. In addition, $r(A)$ is the complement of $r(M)$ in C.

- Consequently, we need to compute the intersection and the complement of the subspaces in a symbolic vector space.

3 How to incarnate the idea into a program?

In this section, we discuss how to incarnate the idea presented in the previous section as an executable program. To this end, we use the *brnoeth* library implemented in a symbolic computation package (Singular). This library is written for the generation of AG code, namely, an implementation of the Brill-Noether algorithm for solving the Riemann-Roch problem and applications to Algebraic Geometry codes; some functions and algorithms used there are also useful for the generation of QEC code.

3.1 A concrete design of the algorithm

In this section, we illustrate a concrete design of the code to generate QECC. The necessary items are as follows:

- The complement of a subspace W in another subspace V.
- The intersection of two subspaces
- The orthonormal basis of a subspace
- The choice of β_k

3.1.1 Symbolic subroutines to transform classical codes into quantum ones

All we need is to perform calculations of linear algebra in the vector space of symbolic items in a quotient ring $\mathbb{Z}[a]/f(a)$ where $f(a)$ is the minimum polynomial of a.

3.1.2 The computation of the complement of a vector space in another one

Let $A \setminus B$ be the complement of two vector subspaces. Those spaces are represented by two matrices A and B; the rows of these matrices give the basis vectors of the corresponding vector spaces. The computation of $A \setminus B$ is a simple procedure.

- Let B_C be a copy of B.
- Choose a vector in A and add it to B_c and extend it. If the dimension of B_c increases by one, then repeat the same step. If not, try with another vector. This step should be repeated while $\dim(B_c) < \dim(A)$.
- When the condition $\dim B_c = \dim(A)$ is satisfied, B_c is made of the basis vectors of B and the set of added vectors $\{v_1, v_2, ..., v_l\}$. The latter is $A \setminus B$.

3.1.3 The computation of the intersection of two subspaces

It is necessary to obtain the intersection $A \cap B$ for two vector subspaces. A and B are the matrices whose rows represent the basis vectors of the corresponding vector spaces. The algorithm is as follows.

- Solve $x \cdot \begin{pmatrix} A \\ -B \end{pmatrix} = 0$. The solution x is decomposed as $x = [v_A, v_B]$. Then $v_A \cdot A = v_B \cdot B$.
- $v_A \cdot A$ or $\cdot B$ is the intersection.
- Equivalently, we solve

$$\begin{pmatrix} A^T & -B^T \end{pmatrix} x = 0 \tag{24}$$

  where $x = \begin{pmatrix} x_A \\ x_B \end{pmatrix}$ The intersection is $(A^T \cdot x_A)^T$ or $(B^T \cdot x_B)^T$.

In the above, we need the computation of $\ker(M)$ of a matrix M. As we work in $\mathbb{F}_q$ where the numbers are defined as symbols, it is convenient to compute the first syzygy (i.e., the module of relations of the given generators) [Wie06, Eis13, ErH11].

The idea of syzygy is summarized as follows.

> Let R be a ring and let $I = (g_1, g_2, ..., g_n)$ be a R-module. The first syzygy of I, denoted Syzygy(I), consists of all relations among the generators of I, which are given by $\sum_i w_i g_i = 0$.

We assume that I is generated by column vectors of the matrix M. Syzygy(I) is computable with the aid of computer algebra packages, and then we get the basis of $\ker M$ through Syzygy(I)[DGPS24].

3.1.4 The normalization of a vector

For a vector $v \in \mathbb{F}_{p^2}$, it is necessary to compute $v/\|v\|$. It is equivalent to getting a normalization constant that consists of a pair of l and $m \in \mathbb{F}_p$ such that $\|(l + \gamma m)\, v\| = 1$. ($\gamma$ stand for the polynomial basis of $\mathbb{F}_{p^2}/\mathbb{F}_p$.) The existence of such (l, m) is guaranteed by the pigeonhole principle, according to a similar discussion as in Lemma 1 in [DGG13].

3.1.5 The computation of an orthonormal basis

To compute the orthonormal basis, one might apply a recursive procedure.

- Let W be a subspace for which an orthogonal basis is computed.
- Pick up a vector such that $\|v\|$=1. Let $\langle v \rangle$ be a subspace made of v.

- Compute the complement $W \setminus \langle v \rangle$
- For each basis of $W \setminus \langle v \rangle$, project out the component parallel to v:

$$u_i \to u_i - (u_i, v)v \tag{25}$$

- Renew $W \setminus v = \{u_1, u_2, ..., u_n\} \to W$.
- Repeat the above steps until W is exhausted. In each step, we obtain a vector v_j, and $\{v_j\}$ is the desired orthogonal basis.

3.1.6 The choice of β_k

The algorithm is simple: this is a search for pairs (u, v) with $u, v \in \mathbb{F}_p$ such that $(u+av)^{p+1} = -1$.

3.2 An example of QECC generation

In the following, we see the ingredients necessary for the QECC one by one.

3.2.1 Generation of algebraic geometry codes

We work in $\mathbb{F}_{p=13}[x, y]$ and use a polynomial $x^3 + y^2 + y$ to generate an algebraic geometry code. To this end, we can use *brnoeth.lib* implemented in SINGULAR [DGPS24].

Let us use the following script of SINGULAR:

```
LIB "latex.lib";
LIB "brnoeth.lib";
LIB "matrix.lib";
int pchar=13;
ring s=pchar,(x,y),lp;
list HC=Adj_div(x3+y2+y);
print(HC);
HC=NSplaces(1..2,HC);
print(HC);
HC=extcurve(2,HC);
print(HC);
def ER=HC[1][4];
setring ER;
intvec G=5;      // the rational divisor G = 5*HC[3][1]
intvec D=2..9;   // D = sum of the rational places no. 2..9
// let us construct the corresponding evaluation AG code :
matrix C=AGcode_L(G,D,HC);
```

This script defines a polynomial $x^3 + y^2 + y$ in $\mathbb{F}_{13}[x, y]$, the divisor G, and the rational places D. Then it computes the algebraic geometry code with divisors G and D through Brill-Noether algorithm. (For details of divisors and places, see the manual and the relevant libraries of SINGULAR.)

The vector basis C of the algebraic geometry code consists of the row vectors of the following matrix.

$$C = \begin{pmatrix} -4 & 3 & 1 & -4 & 3 & 1 & 0 & 0 \\ -4 & -4 & -4 & 3 & 3 & 3 & 0 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -3 & 4 & -1 & -3 & 4 & -1 & 0 & 0 \\ -3 & -1 & 4 & -1 & 4 & -3 & 0 & 0 \end{pmatrix}$$

The entries of C belong to the ring $\mathbb{Z}_{13}[a]/(a^2 + a + 5)$, which is an extension of $\mathbb{Z}_{13}$. In the following, the symbolic computations are carried out in $\mathbb{Z}_{13}[a]/(a^2 + a + 5)$.

3.2.2 Generation of quantum error correction codes

Let C^p be a matrix defined by $C^p_{i,j} = (C_{i,j})^p$. It is given by

$$C^p = \begin{pmatrix} -4 & 3 & 1 & -4 & 3 & 1 & 0 & 0 \\ -4 & -4 & -4 & 3 & 3 & 3 & 0 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -3 & 4 & -1 & -3 & 4 & -1 & 0 & 0 \\ -3 & -1 & 4 & -1 & 4 & -3 & 0 & 0 \end{pmatrix}$$

Using C^p, we compute $C_{\perp_H}$ (the vector space h-perpendicular to C). It consists of the rows of the following matrix:

$$C_{\perp_H} = \begin{pmatrix} 4 & -3 & -1 & -4 & 3 & 1 & 0 & 0 \\ 0 & 3 & 4 & 4 & 1 & 0 & 1 & 0 \\ 1 & 0 & 4 & 3 & 4 & 0 & 0 & 1 \end{pmatrix}$$

We compute the intersection of C and $C_{\perp_H}$. Let T_C be a matrix defined by

$$T_C = \begin{pmatrix} C^T & -C^T_{\perp_H} \end{pmatrix}.$$

It is given by

$$T_C = \begin{pmatrix} -4 & -4 & 1 & -3 & -3 & -4 & 0 & -1 \\ 3 & -4 & 1 & 4 & -1 & 3 & -3 & 0 \\ 1 & -4 & 1 & -1 & 4 & 1 & -4 & -4 \\ -4 & 3 & 1 & -3 & -1 & 4 & -4 & -3 \\ 3 & 3 & 1 & 4 & 4 & -3 & -1 & -4 \\ 1 & 3 & 1 & -1 & -3 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

$\mathrm{Ker}(T_C)$ (the left kernel) is the ingredient for the aimed intersection, which is given by

$$\begin{pmatrix} 2 & 0 & 0 & 0 & -4 & 1 & 0 & 0 \end{pmatrix}$$

Using $\mathrm{Ker}(T_C)$, we obtain $r(M)$ (the intersection of C and $C_{\perp_H}$). It consists of the row of the matrix:

$$M = \begin{pmatrix} 4 & -3 & -1 & -4 & 3 & 1 & 0 & 0 \end{pmatrix}$$

We calculate a matrix A in such a way that the union of $r(A)$ and $r(M)$ is equal to C. It is given by:

$$A = \begin{pmatrix} -4 & 3 & 1 & -4 & 3 & 1 & 0 & 0 \\ -4 & -4 & -4 & 3 & 3 & 3 & 0 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -3 & 4 & -1 & -3 & 4 & -1 & 0 & 0 \end{pmatrix}$$

We calculate the subspace $(C_{\perp_H} \setminus r(M))$, which consists of the rows of the following matrix:

$$B_0 = \begin{pmatrix} 0 & 3 & 4 & 4 & 1 & 0 & 1 & 0 \\ 1 & 0 & 4 & 3 & 4 & 0 & 0 & 1 \end{pmatrix}$$

The basis vectors in the above matrix are neither normalized nor orthogonal; they should undergo a Gram-Schmidt process.

Let us apply the Gram-Schmidt process.

- In the first iteration, we make two vectors: the first one v_1 is a vector of length 1 in $r(B_0)$, while the second t_1 is the basis of $r(B_0) \setminus \{v_1\}$. We use the first row of B_0 as v_1. They are given by

$$v_1 = \begin{pmatrix} (3a-1) & 3 & -a & -(4a-1) & -(a+3) & 0 & 1 & (3a-1) \end{pmatrix}.$$

$$t_1 = \begin{pmatrix} (a-4) & (2a-2) & -(2a-3) & -(3a+6) & -(4a-5) & 0 & (5a-5) & (a-4) \end{pmatrix}.$$

- In the second iteration, we again apply the Gram-Schmidt process to $r(W) = r(B_0) \setminus v_1 = \{t_1\}$ and obtain two vectors $v_2 = t_1/\|t_1\|$ and $t_2 = 0$.

$$v_2 = \begin{pmatrix} (5a+5) & (4a-3) & -(5a-3) & (3a-2) & (4a+6) & 0 & -(3a+1) & (5a+5) \end{pmatrix}.$$

$$t_2 = \begin{pmatrix} 0,0,0,0,0,0,0,0 \end{pmatrix}.$$

- Now we arrive at the vector space $r(W) = \{t_2\} = (0,0,0,0,0,0,0,0)$. We finish the calculation.

- Consequently, $r(B)$ consists of $\{v_1, v_2\}$, i.e., the row vectors of the following matrix:

$$B = \begin{pmatrix} (3a-1) & 3 & -a & -(4a-1) & -(a+3) & 0 & 1 & (3a-1) \\ (5a+5) & (4a-3) & -(5a-3) & (3a-2) & (4a+6) & 0 & -(3a+1) & (5a+5) \end{pmatrix}.$$

We seek an element β such that $\beta\bar{\beta} = -1$; this element is placed on the diagonal part of the sub-matrix in 23. There are several candidates for such β.

$$(4a+1), (4a+3), (-6a+3), (-6a+4), (-a+4),$$
$$5, (a+5), -5, (-a-5), (a-4),$$
$$(6a-4), (6a-3), (-4a-3), (-4a-1),$$

Now we obtain the matrix G of (23):

$$G = \begin{pmatrix} 4 & -3 & -1 & -4 & 3 & 1 & 0 & 0 & 0 & 0 \\ -4 & 3 & 1 & -4 & 3 & 1 & 0 & 0 & 0 & 0 \\ -4 & -4 & -4 & 3 & 3 & 3 & 0 & -1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ -3 & 4 & -1 & -3 & 4 & -1 & 0 & 0 & 0 & 0 \\ 3a-1 & 3 & -a & 1-4a & -a-3 & 0 & 1 & 3a-1 & 4a+1 & 0 \\ 5a+5 & 4a-3 & 3-5a & 3a-2 & 4a+6 & 0 & -3a-1 & 5a+5 & 0 & 4a+1 \end{pmatrix}$$

Recall that the entries of G belong to $\mathbb{F}_{p^2}$ and are generated by the map $\phi(x, y) = x + ay$ with $x, y \in \mathbb{F}_p$. Using the inverse map of ϕ, we get two vectors $r_1, r_2 \in \mathbb{F}_p$ from a row vector r of G. The concatenation (r_1, r_2) stands for a row vector of a matrix representing the corresponding QECC code. Concerning the final result, see the next page.

$$\text{QECC row vectors :} \left[\begin{array}{cccccccccc|cccccccccc}
4 & -3 & -1 & -4 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-4 & 3 & 1 & -4 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-4 & -4 & -4 & 3 & 3 & 3 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-3 & 4 & -1 & -3 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & 1 & -3 & 0 & 1 & -1 & 1 & 0 & 3 & 0 & -1 & -4 & -1 & 0 & 0 & 3 & 4 & 0 \\
5 & -3 & 3 & -2 & 6 & 0 & -1 & 5 & 0 & 1 & 5 & 4 & -5 & 3 & 4 & 0 & -3 & 5 & 0 & 4
\end{array}\right]$$

4 Concluding remarks

In this article, we reviewed the basic idea concerning quantum error correction codes and demonstrated how to generate a type of QEC in concrete terms. The programs used in the study are available at `https://github.com/kikuchiichio/20251119/`

References

[AK00] Alexei Ashikhmin and Emanuel Knill. Nonbinary quantum stabilizer codes, 2000.

[BCL23] Elena Berardini, Alain Couvreur, and Grégoire Lecerf. A proof of the brill-noether method from scratch. *ACM Communications in Computer Algebra*, 57(4):200–229, December 2023.

[CF02] Antonio Campillo and J Farrán. Symbolic hamburger-noether expressions of plane curves and applications to ag codes. *Mathematics of computation*, 71(240):1759–1780, 2002.

[CRSS97] A. R. Calderbank, E. M Rains, P. W. Shor, and N. J. A. Sloane. Quantum error correction via codes over gf(4), 1997.

[DGG13] Akshay Degwekar, Kenza Guenda, and T. Aaron Gulliver. Extending construction x for quantum error-correcting codes, 2013.

[DGPS24] Wolfram Decker, Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann. SINGULAR 4-4-0 — A computer algebra system for polynomial computations. `http://www.singular.uni-kl.de`, 2024.

[Eis13] David Eisenbud. *Commutative algebra: with a view toward algebraic geometry*, volume 150. Springer Science & Business Media, 2013.

[ErH11] Viviana Ene and J rgen Herzog. *Grobner bases in commutative algebra*, volume 130. American Mathematical Soc., 2011.

[GP17] Giuliano Gadioli La Guardia and Francisco Revson F. Pereira. Good and asymptotically good quantum codes derived from algebraic geometry codes, 2017.

[Gra20] Markus Grassl. Algebraic quantum codes: linking quantum mechanics and discrete mathematics. *International Journal of Computer Mathematics: Computer Systems Theory*, 6(4):243–259, December 2020.

[Gre17] Gert-Martin Greuel. Singularities in positive characteristic, 2017.

[KKKS05] Avanti Ketkar, Andreas Klappenecker, Santosh Kumar, and Pradeep Kiran Sarvepalli. Nonbinary stabilizer codes over finite fields, 2005.

[MTT16] Carlos Munuera, Wanderson Tenório, and Fernando Torres. Quantum error-correcting codes from algebraic geometry codes of castle type. *Quantum Information Processing*, 15(10):4071–4088, July 2016.

[Wie06] Roger Wiegand. Communications-what is a syzygy? *Notices of the American Mathematical Society*, 53(4):456–457, 2006.