

# Searches for Optimal Quaternary Hermitian Linear Complementary Dual Codes for Entanglement-Assisted Quantum Error Correction

Maysara Al Jumaily

Department of Computer Science  
Brock University

1812 Sir Isaac Brock Way  
St. Catharines, ON L2S 3A1, Canada  
amaysara@brocku.ca

Sheridan Houghten

Department of Computer Science  
Brock University

1812 Sir Isaac Brock Way  
St. Catharines, ON L2S 3A1, Canada  
shoughten@brocku.ca

**Abstract**—We present the results of a series of searches for Hermitian linear LCD codes for entanglement-assisted quantum error correction. Such codes are *optimal* if they have the largest possible minimum distance given their length and dimension. We present two optimal codes: a  $[17, 11, 5]_4$  and a  $[17, 13, 3]_4$  code, which can be transformed into entanglement-assisted codes. We also report ten codes that improved the lower bounds used in our searches, presenting  $[22, 9, 9]_4$ ,  $[22, 12, 7]_4$ ,  $[23, 8, 11]_4$ ,  $[23, 10, 9]_4$ ,  $[23, 13, 7]_4$ ,  $[24, 10, 10]_4$ ,  $[24, 11, 9]_4$ ,  $[24, 14, 7]_4$ ,  $[25, 12, 9]_4$ , and  $[25, 13, 8]_4$  codes. Finally, we amend some of the parameters in related literature, and share our own Java program for performing such searches.

**Index Terms**—Quantum codes, optimal Hermitian LCD codes, entanglement-assisted quaternary error correction

## I. INTRODUCTION

Linear complementary dual (LCD) codes were discovered by Massey in 1992 [1]. Now, researchers are interested in finding Hermitian LCD codes because they can be translated into quantum codes, in particular entanglement-assisted quantum codes. A classical linear code  $C$  can be imported as a standard quantum error correcting code (QECC) provided that  $C$  is self-orthogonal (i.e.,  $C \subset C^\perp$ ) or self-dual (i.e.,  $C = C^\perp$ ). An entanglement-assisted quantum error correcting code (EAQECC) is a generalization of a standard QECC. It uses *ebits* to bypass the self-orthogonality/duality constraint, where an ebit is a two-level maximally entangled quantum system that is shared between the sender and receiver prior to the beginning of communication [2], [3]. They are used to translate a noncommuting quantum group of generators to a commutative group to allow the quantum error correction techniques to operate.

## II. BACKGROUND

Let  $C = [n, k, d]_q$  be a linear code of dimension  $k$  over the field  $\mathbb{F}_q^n$  where  $q$  is the cardinality of the symbols and  $n$  the length of each codeword and  $d$  is the minimum Hamming

distance between any codewords in  $C$ . The *Hamming weight* of a codeword is the number of its nonzero elements. The all-zero vector, denoted by  $\mathbf{0}_n$ , is a codeword by default in any code  $C$ . The *minimum* Hamming distance is the smallest weight of any codeword in  $C$ . Let  $G_{k \times n}$  and  $H_{(n-k) \times n}$  be the generator matrix and parity check matrix, respectively.

We will use the notation  $M^T$  and  $M^\dagger$  to denote the transpose and conjugate transpose of the matrix  $M$ , respectively. The *weight enumerator* is a polynomial function which contains the Hamming weights of all possible codeword in a linear code  $C$ .

**Definition 1:** The weight enumerator of a linear code  $C$  over  $\mathbb{F}_q^n$  is given by

$$w(x)_C = A_0 + A_1x + \dots + A_{n-1}x^{n-1} + A_nx^n,$$

where  $A_i$  is the number of codewords of  $C$  whose Hamming weight is  $i$ .

A binary linear code of length  $n$  is defined over  $\mathbb{F}_2^n$ .

A quaternary linear code of length  $n$  is defined over  $\mathbb{F}_4^n$  and has the following elements:  $\{0, 1, \omega, \bar{\omega}\}$ . The *conjugation* of an element  $a \in \mathbb{F}_{q^2}$  is  $a^2$ , and is denoted as  $a^\dagger$ .

**Definition 2:** The dual of a linear code  $C$  over  $\mathbb{F}_q^n$  is denoted by  $C^\perp$  and defined as

$$C^\perp = \{v \in \mathbb{F}_q^n \mid \langle v, w \rangle = 0, \forall w \in C\}, \quad (1)$$

where  $v = [v_1 \ v_2 \ \dots \ v_n]$ ,  $w = [w_1 \ w_2 \ \dots \ w_n]$  and  $\langle v, w \rangle = \sum_{i=1}^n v_i \cdot w_i$ .

**Definition 3:** The *Hermitian dual* of a linear code  $C$  over  $\mathbb{F}_{q^2}^n$  is denoted by  $C^{\perp_H}$  and defined as

$$C^{\perp_H} = \{v \in \mathbb{F}_{q^2}^n \mid \langle v, w \rangle_H = 0, \forall w \in C\}, \quad (2)$$

where  $v = [v_1 \ v_2 \ \dots \ v_n]$ ,  $w = [w_1 \ w_2 \ \dots \ w_n]$  and  $\langle v, w \rangle_H = \sum_{i=1}^n v_i \cdot w_i^\dagger$ .

**Definition 4:** A code  $C$  over  $\mathbb{F}_{q^2}^n$  is *Hermitian LCD* if  $C \cap C^{\perp_H} = \{\mathbf{0}_n\}$ .

Let  $C_q = \llbracket n, k, d; c \rrbracket$  be an entanglement-assisted linear quantum code of dimension  $k$  over the field  $\mathbb{F}_q^n$  where  $q \in \{2, 4\}$  is the cardinality of the symbols and  $n$  the length of

each codeword,  $d$  is the minimum Hamming distance between any pair of codewords in  $C_q$  and  $c$  is the number of ebits required. An EAQECC is constructed from binary or quaternary codes. Let us mention Corollary 3.6.1 [3]:

*Corollary 1:* An EAQECC can be generated from two classical codes  $C_1, C_2 \in \mathbb{F}_2^n$  where  $C_1 = [n, k_1, d_1]$  and  $C_2 = [n, k_2, d_2]$ . Let  $H_1$  and  $H_2$  denote the parity check matrices for  $C_1$  and  $C_2$ , respectively, then an entanglement-assisted quantum code  $C_q$  with parameters  $[[n, k_1 + k_2 - n + c, \min(d_1, d_2); c]]$  can be obtained where  $c = \text{RANK}(H_1 H_2^T)$ .

Furthermore, let us also mention Corollary 3.6.2 from article [3] which relies two other references:

*Corollary 2:* an EAQECC  $C_q$  can be generated from a classical quaternary linear code  $C = [n, k, d]_4$  with parity check matrix  $H$  with parameters  $[[n, 2k - n + c, d; c]]$ , where  $c = \text{RANK}(HH^\dagger)$  [4], [5].

In this paper, we will only focus on quaternary Hermitian LCD codes because they generate entanglement-assisted quantum codes with parameters  $[[n, 2k - n + c, d; n - k]]$ .

*Definition 5:* A quaternary Hermitian LCD code  $[n, k, d]_4$  is said to be *optimal* if no Hermitian LCD code with parameters  $[n, k, d']_4$  exists with  $d' > d$ .

### III. METHODOLOGY FOR CONSTRUCTION OF OPTIMAL HERMITIAN LCD CODES

Most previous work, for example articles [6]–[11] used <http://codetables.de/> [12] and/or MAGMA [13] to obtain their results. In our work, we use a similar construction to that presented in [14]. Every  $[n, k, d]_4$  code is equivalent to one with a generator matrix in standard form:  $G = [I_{k \times k} \mid P_{k \times (n-k)}]_{k \times n}$ , where  $I$  is the identity matrix and  $P$  is a  $k \times (n - k)$  matrix [14].

We can assume the following characteristics for the submatrix  $P$ , where  $r_i$  denotes the  $i^{\text{th}}$  row in  $P$ :

- The top row is a fixed row vector of the form

$$\underbrace{[1 \ 0 \ \dots \ 0]}_{n-k-d+1} \underbrace{[1 \ 1 \ \dots \ 1]}_{d-1}. \quad (3)$$

- The first nonzero digit from the left is 1 for all rows (*i.e.*, neither  $\omega$  nor  $\bar{\omega}$ ).
- $\forall r_i : \text{WT}(r_i) \geq d - 1$ . Note that the identity matrix contributes 1 to the overall weight of each row of the generator matrix, bringing the overall weight of each row of the generator matrix to  $\geq d$ .
- We use lexicographic ordering so that the rows are sorted in ascending order when  $d \geq 3$ :

$$r_1 < r_2 < \dots < r_k, \quad (4)$$

or the following lexicographic ordering if  $d \not\geq 3$ :

$$r_1 \leq r_2 \leq \dots \leq r_k. \quad (5)$$

Note however that we do not currently employ Equation 5 restriction in our work.

The approach was to construct the submatrix  $P$  and the generator matrix  $G$  dynamically in terms of the rows and columns [14]. It started with having the top row fixed. Then, for each iteration  $m = 2, \dots, k - 1$ , the number of rows

increased as well as the columns, appropriately. The generator matrix was constructed as follows:

$$\left[ \begin{array}{c|c} I_{m \times m} & \begin{matrix} r_1 \\ \vdots \\ r_m \end{matrix} \end{array} \right]. \quad (6)$$

Furthermore, at each step, the constructed code was in the form  $[n + m - k, m]_4$  and the minimum distance at least  $d$ . Our construction is slightly different in that the length  $n$  of the code is fixed and the generator matrix will dynamically add rows that satisfy the orthogonality constraint. Also, each row that is added, along with all linear combinations, must have a minimum weight of least  $d$ .

#### A. The Programming Implementation

The source code of the program discussed is freely available at: <https://github.com/aljumaily/MScThesis>. Our implementation uses Java as the programming language to implement an efficient approach to find optimal quaternary Hermitian LCD codes. The implementation uses only Java's 8 standard libraries and no external packages. The program is designed to use a long, 64-bit data type, to represent the codewords. Since each quaternary value requires two bits to be represented, the maximum value for  $n$ , the length of the code, is 32. The quaternary values are mapped to 2-bit binary values:  $0_4 \mapsto 00_2$ ,  $1_4 \mapsto 01_2$ ,  $\omega_4 \mapsto 10_2$  and  $\bar{\omega}_4 \mapsto 11_2$ . The program implements the Bareiss algorithm [15], a fraction-free method for computing the determinant of a matrix. A pseudocode is found in Algorithm 1. Furthermore, the main recursive engine of our implementation is presented in Algorithm 2. MATLAB R2021a was used for the sake of confirming the matrices found are indeed invertible and have full rank<sup>1</sup>.

### IV. RESULTS

Our computational searches revealed that  $[17, 11, 5]_4$  and  $[17, 13, 3]_4$  are optimal Hermitian LCD codes. These results were obtained by using our program to perform an exhaustive search for  $[17, 11, 6]_4$  and  $[17, 13, 4]_4$  codes, finding that no such Hermitian LCD codes exist. This means there exist optimal EAQECCs in the form of  $[[17, 11, 5; 6]]$  and  $[[17, 13, 3; 4]]$ . The generator matrices and weight enumerators of the two codes are presented next.

We performed searches for several other parameters and improved ten lower bounds relative to the literature [16]. We use the notation  $[n, k, d_{\min} - d_{\max}]_4$  to denote a quaternary code for which the lower bound on the largest minimum distance is  $d_{\min}$  and the corresponding upper bound is  $d_{\max}$ . In each of the following cases, our searches found codes with a minimum distance that increased the then-used lower bound by 1:  $[22, 9, 9 - 10]_4$ ,  $[22, 12, 7 - 8]_4$ ,  $[23, 8, 11 - 12]_4$ ,  $[23, 10, 9 - 10]_4$ ,  $[23, 13, 7 - 8]_4$ ,  $[24, 10, 10 - 11]_4$ ,  $[24, 11, 9 - 10]_4$ ,  $[24, 14, 7 - 8]_4$ ,  $[25, 12, 9 - 10]_4$  and  $[25, 13, 8 - 9]_4$ . Codes attaining these lower bounds are also presented in Section V.

<sup>1</sup>Matlab's `ctranspose(M)` (or equivalently `M'`) function that compute the conjugate transpose of the matrix  $M$  does not work as expected when  $M$  is a Galois field matrix. This is discussed in-depth in [16] section 8.2.6.

$$G_{[17, 11, 5]_4} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & \omega \\ 0 & 1 & 1 & 0 & \omega & 1 \\ 0 & 1 & 1 & \omega & 1 & 0 \\ 0 & 1 & \omega & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & \bar{\omega} \\ 1 & 0 & 1 & 0 & \omega & \bar{\omega} \\ 1 & 0 & 1 & \omega & \bar{\omega} & 1 \\ 1 & 0 & \omega & 1 & \bar{\omega} & 0 \\ 1 & 1 & 0 & \bar{\omega} & \omega & \omega \\ 1 & 1 & 1 & 0 & \bar{\omega} & 0 \end{bmatrix} \quad (7)$$

$$G_{[17, 13, 3]_4} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & \omega \\ 0 & 0 & 1 & \bar{\omega} \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & \omega \\ 0 & 1 & 0 & \bar{\omega} \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & \omega \\ 0 & 1 & 1 & \bar{\omega} \\ 0 & 1 & \omega & 0 \\ 0 & 1 & \omega & 1 \\ 0 & 1 & \bar{\omega} & 0 \end{bmatrix} \quad (8)$$

$$\text{WT}(x)_{[17, 11, 5]_4} = 171151 + 471x^5 + 2,415x^6 + 9,609x^7 + 39,363x^8 + 117,654x^9 + 279,414x^{10} + 535,122x^{11} + 802,842x^{12} + 927,507x^{13} + 793,131x^{14} + 476,373x^{15} + 178,962x^{16} + 31,440x^{17}$$

$$\text{WT}(x)_{[17, 13, 3]_4} = 171331 + 267x^3 + 2,385x^4 + 16,314x^5 + 91,278x^6 + 391,545x^7 + 1,318,581x^8 + 3,518,460x^9 + 7,388,436x^{10} + 12,091,725x^{11} + 15,110,175x^{12} + 13,952,682x^{13} + 8,966,910x^{14} + 3,587,535x^{15} + 672,570x^{16}$$

TABLE I  
SELECTED BOUNDS ON THE LARGEST MINIMUM DISTANCE OF  
QUATERNARY HERMITIAN LCD CODES. BOLD LOWER ENDPOINTS OR  
EXACT VALUES IDENTIFY DISTANCES ATTAINED BY OUR CONSTRUCTIONS.  
UNBOLDED ENTRIES ARE LITERATURE COMPARISONS.

| $n \backslash k$ | 8            | 9           | 10           | 11          | 12          | 13         | 14  |
|------------------|--------------|-------------|--------------|-------------|-------------|------------|-----|
| 17               |              |             |              | <b>5</b>    |             | <b>3</b>   |     |
| 22               |              | <b>9-10</b> | 9            | 8-9         | 7-8         |            |     |
| 23               | <b>11-12</b> |             | <b>9-10</b>  |             |             | 7-8        |     |
| 24               |              |             | <b>10-11</b> | <b>9-10</b> |             |            | 7-8 |
| 25               |              |             |              |             | <b>9-10</b> | <b>8-9</b> |     |

Table I distinguishes our constructions from literature-only comparisons. The bounds for our twelve parameter pairs agree with Table 6 of [17]. The unbolded entries are the interval 8–9 for  $(n, k) = (22, 11)$  from [17] and the exact value 9 for  $(22, 10)$ , obtained by combining the upper bound in [17] with the construction in Example 2 of [18]. Neither unbolded entry is counted among the contributions documented here. This is a dated literature snapshot, not a claim of universally latest bounds.

#### A. Confirmation and Updating of other Parameters

We will now focus on the optimal parameters given in references [6]–[11], [14]. We performed searches for all optimal parameters given for  $2 \leq k \leq 14$  and  $2 \leq n \leq 14$  using our implementation and matched almost all of the parameters provided. However, some entries are inconsistent from one reference to another while another entry must be amended due to incomplete information. The following entries must be amended:  $[12, 7, 5]_4$  from [7] should have 4 as its largest minimum distance, as [8] mentioned this;  $[13, 11, 3]_4$  from [9] should have a largest minimum distance of 2, as what [7], [8] concluded;  $[14, 10, 4]_4$  from [7]–[9] should be  $[14, 10, 3]_4$ ; and finally  $[14, 11, 4]_4$  from [9] should be  $[14, 11, 3]_4$  because of both [7], [8] established this result.

Finally, the optimal distance of 4 for  $[14, 10, 4]_4$  was first mentioned in [7] but neither it nor the papers it cited included a generator matrix of a  $[14, 10, 4]_4$  code; this same value was listed in [8], [9] but again, without a generator matrix. Our search found that there exists a  $[14, 10, 4]_4$  quaternary code, however it is *not* a Hermitian LCD code. As a result, we conclude that  $[14, 10, 3]_4$  is an optimal Hermitian LCD code. Our program also matched the modified largest minimum distance for all these parameters.

#### V. NEW CODES OBTAINED

In this section, we share where to access the generator matrices and weight enumerators for the twelve constructions reported: two optimal codes and ten codes that improved the lower bounds used at that time. All can be accessed through [16] in section 9. For the sake of completeness, we share the generator matrix and weight enumerator for a  $[23, 8, 11]_4$  code, which has the largest minimum distance of all codes found.

#### VI. CONCLUSIONS AND FUTURE WORK

In this paper we report our computer searches for optimal quaternary Hermitian LCD codes. Our contributions comprise two optimal constructions and ten lower-bound improvements, whose distances agree with the reviewed September 2026 literature bounds. The literature-only entries in Table I are not additional contributions of our searches. We have also confirmed and/or amended several parameters given in the literature.

Another major contribution is the creation of a Java program specifically designed for finding such codes, and that is available for other researchers to use. The program can be enhanced in the future to allow for executing parameters with  $d < 3$ , and to include a graphical user interface for ease of use. Other suggested improvements can be found on the GitHub page where the program is hosted and we share some: implementing binary LCD and quaternary LCD (quaternary Hermitian LCD is what this program implemented);

$$G_{[23,8,11]_4} = \left[ \begin{array}{cccccccc|cccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & \omega & \omega & \omega & \bar{\omega} & \bar{\omega} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & 1 & \omega & 1 & \omega & \bar{\omega} & 0 & 1 & \bar{\omega} & 0 & \omega \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \bar{\omega} & 1 & \bar{\omega} & \omega & 1 & \bar{\omega} & \omega & \bar{\omega} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & \bar{\omega} & \omega & \bar{\omega} & \bar{\omega} & 1 & \omega \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & 0 & 0 & 1 & \omega & \bar{\omega} & 1 & \omega & 1 & \bar{\omega} & 0 & \omega & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & \omega & \bar{\omega} & 1 & \omega & 0 & \omega & \bar{\omega} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \omega & 1 & \omega & \omega & 0 & \omega & 1 & \omega & 1 & \omega & 0 \end{array} \right] \quad (9)$$

$$\begin{aligned} \text{WT}(x)_{[23, 8, 11]_4} = & 1 + 486x^{11} + 741x^{12} + 1,560x^{13} + 3,423x^{14} + 6,375x^{15} + 9,813x^{16} \\ & + 12,300x^{17} + 12,366x^{18} + 9,480x^{19} + 5,925x^{20} + 2,388x^{21} + 627x^{22} + 51x^{23} \end{aligned} \quad (10)$$

adding Graphical User Interface to make the program more user-friendly instead of using command prompt; including a multithreading option that is used to check whether a vector is linearly orthogonal to other codewords; lastly, a more significant

feature is using 128-bit vectors, which would be a combination of two longs to represent a single vector. That way, we can increase the maximum n value to about 60.

APPENDIX  
ALGORITHMS

The determinant calculation and recursive search are summarized in Algorithms 1 and 2, respectively.

**Algorithm 1:** Bareiss Algorithm which takes in a quaternary  $n \times n$  matrix represented by binary values and return its determinant which can either be 0, 1,  $\omega$  or  $\bar{\omega}$  or 00<sub>2</sub>, 01<sub>2</sub>, 10<sub>2</sub> and 11<sub>2</sub> in binary, respectively

**Input** :  $M_{n \times n}$ : A square matrix to find the determinant of.

**Output:**  $\det (M_{n \times n}) \in \{0, 1, 2, 3\}$

```

1 Function bareissAlgorithm( $M_{n \times n}$ ):
2    $pivot \leftarrow 1$ 
3   for  $k \leftarrow 0$  to  $n - 2$  do
4     if  $M[k][k] = 0$  then // Division-by-zero encountered
5       Avoid division-by-zero by swapping current row
        with another one
7        $isFixed \leftarrow \text{false}$ 
8       for  $r \leftarrow k + 1$  to  $n - 1$  do
9         if  $M[r][k] \neq 0$  then
10          swap row  $k$  with  $r$ 
11           $isFixed \leftarrow \text{true}$ 
12          break
13       end
14       end
15       if  $isFixed = \text{false}$  then // We still have
        division-by-zero
16         matrix is not invertible
17         return 0
18       end
19       for  $i \leftarrow k + 1$  to  $n - 1$  do
20         for  $j \leftarrow k + 1$  to  $n - 1$  do
21            $result \leftarrow$ 
22             
$$\frac{M[i][j] \times M[k][k] - M[i][k] \times M[k][j]}{pivot}$$

23            $M[i][j] \leftarrow result$ 
24         end
25       end
26        $pivot \leftarrow M[k][k]$ 
27   end
28   return  $M[n - 1][n - 1] \in \{1, 2, 3\}$ 
29 End Function

```

**Algorithm 2:** The main engine of the program that starts the search to conclude whether or not a  $[n, k, d]_4$  exists

**Data** :  $n, k, d, G_{k \times n}$

**Input** :  $r$ : The current row of  $G$  being filled and it is zero-based,  $0 \leq r \leq k$ .

$vGen$ : The vector generator that generates potential codewords to insert in  $G$ .

**Output :** true if  $[n, k, d]_4$  code exists, false otherwise

```

1 Function isCodeValid( $r$ ,  $vGen$ ):
2   if  $r \geq k$  then                                     // Base case
3     if  $\det(G \times G^\dagger) \neq 0$  then                // Valid generator matrix
4       return true
5     else
6        $G[r-1] \leftarrow 0$ 
7       return false
8     end
9   end
10  while true do
11     $vector \leftarrow vGen.nextVector()$ 
12    // Check if vector generator cannot generate new
    vectors
13    if ! $vGen.isValid()$  then
14      return false
15    end
16    // Check if vector is orthogonal to  $4^r$  codewords found
17    if isOrthogonal(vector) then
18       $G[r] \leftarrow vector$ 
19      // Create a new instance of the vector generator
      and start at the value of vector + 1
20       $x \leftarrow new\ vGen(vector + 1)$ 
21      if isCodeValid( $r + 1, x$ ) then                // Recursive
        call
22        return true
23      end
24    end
25  end
26 End Function

```

## REFERENCES

- [1] J. Massey, Linear codes with complementary duals, *Discrete Mathematics* 106-107 (1992), pp. 337-342, DOI: 10.1016/0012-365X(92)90563-U.

- [2] S. Taghavi *et al.*, Optimized entanglement-assisted quantum error correction, *Physical Review A* 82.4 (Oct. 2010).
- [3] M. Wilde, Quantum Coding with Entanglement (2008). doi:

10.48550/arXiv.0806.4214

- [4] T. Brun *et al.*, Correcting Quantum Errors with Entanglement, *Science* 314.5798 (2006), 10.1126/science.1131563.
- [5] T. Brun *et al.*, Catalytic quantum error correction, quant-ph/0608027.
- [6] C. Lai *et al.*, Duality in Entanglement-Assisted Quantum Error Correction, *IEEE Transactions on Information Theory* 59.6 (2013), pp. 4020-4024, doi: 10.1109/TIT.2013.2246274.
- [7] L. Lu *et al.*, Maximal entanglement entanglement-assisted quantum codes constructed from linear codes, *Quantum Information Processing* 14.1 (Jan. 2015), pp. 165-182, doi: 10.1007/s11128-014-0830-y.
- [8] C. Lai *et al.*, Linear Programming Bounds for Entanglement-Assisted Quantum Error-Correcting Codes by Split Weight Enumerators, *IEEE Transactions on Information Theory* 64.1 (2018), pp. 622-639, doi: 10.1109/TIT.2017.2711601.
- [9] L. Lu *et al.*, Optimal Quaternary Hermitian LCD codes, arXiv: 2010.10166 [cs.IT].
- [10] M. Harada, Construction of binary LCD codes, ternary LCD codes and quaternary Hermitian LCD codes. *Designs, Codes and Cryptography*, doi: 10.1007/s10623-021-00916-1.
- [11] A. 11 *et al.*, New constructions of entanglement-assisted quantum codes. *Cryptography and Communications*, pp. 2295-2312, doi: 10.1007/s12095-021-00499-7.
- [12] M. Grassl, Code Tables: Bounds on the parameters of various types of codes, <http://codetables.de/>.
- [13] W. Bosma *et al.*, The 13 Algebra System I: The User Language, *Journal of Symbolic Computation* 24.3 (1997), pp. 235-265, doi: 10.1006/jsco.1996.0125.
- [14] M. Harada, Some optimal entanglement-assisted quantum codes constructed from quaternary Hermitian linear complementary dual codes, *International Journal of Quantum Information* 17.07 (2019), doi: 10.1142/S0219749919500539.
- [15] E. Bareiss, Sylvester's identity and multistep integer-preserving Gaussian elimination, *Mathematics of computation* 22.103 (1968), pp. 565-578.
- [16] M. Al Jumaily, Optimal Quaternary Hermitian Linear Complementary Dual Codes for Entanglement-Assisted Quantum Error Correction, <http://hdl.handle.net/10464/15405>
- [17] M. Araya and M. Harada, On the minimum weights of quaternary Hermitian LCD codes, *Cryptography and Communications* 16 (2024), pp. 1539-1558, Table 6, doi: 10.1007/s12095-024-00733-y.
- [18] K. Abdukhalikov and D. Ho, On the equivalence between additive and linear codes, *arXiv:2603.15071v2* (2026), Example 2, <https://arxiv.org/abs/2603.15071v2>.