Toward Quantum CSS-T Codes from Sparse Matrices

Eduardo Camps-Moreno, Hiram H. López, Gretchen L. Matthews, & Emily McMillon
Department of Mathematics
Virginia Tech
Blacksburg, VA, USA
Email: {e.camps, hhlopez, gmatthews, emcmillon}@vt.edu

***Abstract*—CSS-T codes were recently introduced as quantum error-correcting codes that respect a transversal gate. A CSS-T code depends on a pair (C_1, C_2) of binary linear codes C_1 and C_2 that satisfy certain conditions. We prove that C_1 and C_2 form a CSS-T pair if and only if $C_2 \subset \mathrm{Hull}(C_1) \cap \mathrm{Hull}(C_1^2)$, where the hull of a code is the intersection of the code with its dual. We show that if (C_1, C_2) is a CSS-T pair, and the code C_2 is degenerated on $\{i\}$, meaning that the i^{th}-entry is zero for all the elements in C_2, then the pair of punctured codes $(C_1|_i, C_2|_i)$ is also a CSS-T pair. Finally, we provide Magma code based on our results and quasi-cyclic codes as a step toward finding quantum LDPC or LDGM CSS-T codes computationally.**

I. Introduction

Since the 1990s, it has been known that linear codes C_1 and C_2 with $C_2 \subseteq C_1$ may be used to define a quantum stabilizer code $Q(C_1, C_2)$ called a CSS code, based on the work of Calderbank and Shor [1] and Steane [2]. Stabilizer codes are important for quantum error correction, protecting quantum information against errors induced by noise and decoherence, and fault-tolerant quantum computing. It is desirable that quantum error-correcting codes enable gates to be implemented transversally, since transversal gates act independently on the qubits to impede the propagation of errors. Codes with this property are called CSS-T codes, as introduced in [3]. More formally, the code $Q(C_1, C_2)$ is a CSS-T code provided C_2 is an even weight code, meaning all of its codewords have even weight, and for each codeword $c \in C_2$, the shortening of $C_1^\perp$ with respect to the support of c is self-dual. Reed-Muller codes have been employed to define CSS-T codes [4], and the mathematical concepts defining binary CSS-T codes have been considered for larger alphabets [5]. In [6], it was demonstrated that CSS-T codes form a poset, providing tools to study maximal CSS-T codes. Even so, CSS-T codes (especially those satisfying other desirable properties) are elusive and determining asymptotically good families of CSS-T codes is an open question.

Another important family of quantum error-correcting codes is formed by the quantum low-density parity-check (LDPC) codes [7], [8] which are CSS codes based on sparse parity-check matrices. In recent work, such codes have been demonstrated for constant-overhead fault-tolerant quantum computation [9] and to rival surface codes [10], which were introduced in [11]. Additional evidence of the capabilities of quantum LDPC codes may be found in [12], [13], [14], [15], [16].

In this paper, we consider CSS-T codes defined by sparse matrices. We demonstrate a step towards finding a source of such codes by providing an equivalent characterization of CSS-T codes and considering a construction of LDPC codes that have efficient encoding [17]. This paper is organized as follows. Section II reviews the necessary background. Section III includes a useful characterization of binary CSS-T codes in terms of hulls and relative hulls of codes. In Section V, we consider the application to sparse matrix codes. Examples are provided in Section VI. A conclusion is given in Section VII.

II. Preliminaries

In this section, we review the foundation on which our results will be built and set the notation to be used later. We use the standard notation from coding theory. All codes considered in this paper are binary linear codes. The finite field with two elements is $\mathbb{F}_2 := \{0, 1\}$, and $\mathcal{A}^{m\times n}$ is the set of all $m \times n$ matrices whose entries are elements of a set $\mathcal{A}$. Given a matrix $A \in \mathbb{F}_2^{m\times n}$, its transpose is $A^T \in \mathbb{F}_2^{n\times m}$ and its i^{th} row is $\mathrm{Row}_i A$. The standard basis vectors for $\mathbb{F}_2^n$ are $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$, $i \in [n] := \{1, \ldots, n\}$, where the only nonzero entry is in position i, and $\mathbf{1} = \sum_{i\in[n]} e_i \in \mathbb{F}_2^n$ denotes the all ones vector. An $[n, k, d]$ code C is a k-dimensional vector subspace of $\mathbb{F}_2^n$ in which any two distinct codewords (meaning elements of C) differ in at least d positions.

The Schur (also known as the star or pointwise) product of two vectors $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in F_2^n$ is given by

$$x \star y := (x_1y_1, \ldots, x_ny_n).$$

The Schur (also called the star or pointwise) product of binary codes C and C' of length n is denoted and defined by

$$C \star C' := \langle c \star c' : c \in C, c' \in C' \rangle \subseteq \mathbb{F}_2^n,$$

the $\mathbb{F}_2$-span of the Schur products of all codewords of C. We define the square of the code C by $C^2 := C \star C$.

Camps-Moreno and Matthews are partially supported by NSF DMS-2201075. López is partially supported by NSF DMS-2401558. McMillon is supported by NSF DMS-2303380. The authors are partially supported by the Commonwealth Cyber Initiative.

A. Codes defined by sparse matrices

A binary linear code is a low-density parity-check (LDPC) code if it is the null space of a matrix $H \in \mathbb{F}_2^{r\times n}$ with row weight $O(1)$. LDPC codes have superior performance when coupled with iterative message passing algorithms that operate on an associated sparse Tanner graph. A binary linear code is a low-density generator matrix (LDGM) code if it has a generator matrix $G \in \mathbb{F}_2^{k\times n}$ with row weight $O(1)$. LDGM codes facilitate efficient encoding. One may also consider moderate-density parity-check (MDPC) or moderate-density generator matrix (MDGM) codes, which have defining matrices with row weight $O(\sqrt{n\log n})$ or $O(\sqrt{n})$. For LDPC codes, row weights of less than 10 are typically considered [18].

B. CSS codes

We say that Q is $[[n,k,d]]$ to mean a quantum code that encodes k logical qudits into n physical qudits and minimum distance d. Let $C_1 = [n,k_1,d_1]$ and $C_2 = [n,k_2,d_2]$ be two binary codes such that $C_2 \subseteq C_1$. We denote by $Q(C_1,C_2)$ the quantum CSS code defined by C_1 and C_2. Recall that a binary generator matrix for the stabilizer that defines $Q(C_1,C_2)$ can be written as

$$\begin{bmatrix} 0 & H_1 \\ G_2 & 0 \end{bmatrix}, \tag{1}$$

where H_1 is a parity-check matrix for C_1 and G_2 is a generator matrix for C_2. The CSS code $Q(C_1,C_2)$ is a

$$[[n, k_1+k_2-n, \geq \min\{d_1,d_2\}]]$$

quantum code. Notice that the matrix in Expression (1) will be sparse if H_1 is the parity-check matrix of an LDPC code and G_2 is the generator matrix of an LDGM code.

C. Quasi-cyclic codes

We will use quasi-cyclic codes to define the quantum codes of interest. Recall that a code $C \subseteq \mathbb{F}_2^n$ is quasi-cyclic if and only if there exists $l \in [n]$ such that $c = (c_1,\ldots,c_n) \in C$ implies $(c_{n-l+1},\ldots,c_n,c_1,\ldots,c_{n-l}) \in C$, where indices are taken modulo n. Note that the code C is cyclic if it is quasi-cyclic for $l = 1$. Quasi-cyclic codes can be defined by circulant matrices, meaning those of the form

$$\begin{bmatrix} a_1 & a_2 & \ldots & a_n \\ a_n & a_1 & \ldots & a_{n-1} \\ \vdots & \vdots & & \vdots \\ a_2 & a_3 & \ldots & a_1 \end{bmatrix},$$

used as blocks A_{ij} in a generator or parity-check matrix of the form

$$\begin{bmatrix} A_{11} & A_{12} & \ldots & A_{1n} \\ A_{21} & A_{22} & \ldots & A_{2n} \\ \vdots & \vdots & & \vdots \\ A_{t1} & A_{t2} & \ldots & A_{tn} \end{bmatrix}.$$

Quasi-cyclic codes can be compactly described, and, as the next result shows, retain their structure under the Schur product.

Proposition 1. *If C is quasi-cyclic, then C^2 is quasi-cyclic.*

Proof. Suppose $C \in \mathbb{F}_2^n$ is quasi-cyclic and $w \in C^2$. Then $w = \sum_{c,c'\in C} a_{c,c'} c \star c' = \sum_{c,c'\in C} a_{c,c'}(c_1c'_1,\ldots,c_nc'_n)$ where $a_{c,c'} \in \mathbb{F}_2$. Notice that $c,c' \in C$ implies $(c_{n-l+1}c'_{n-l+1},\ldots,c_nc'_n,c_1c'_1,\ldots,c_{n-l}c'_{n-l}) \in C$ since $(c_{n-l+1},\ldots,c_n,c_1,\ldots,c_{n-l}) \in C$ and $(c'_{n-l+1},\ldots,c'_n,c'_1,\ldots,c'_{n-l}) \in C$ which completes the proof. □

III. CSS-T Codes and Hulls

Hulls were first considered by Assmus and Key [19] and have been considered in the context of linearly complementary dual (LCD) codes [20] as well as entanglement-assisted quantum error-correcting codes [17], [21]. In this section, we consider a characterization of CSS-T codes and link it to the hull of a code. First, we recall the definition of a CSS-T code as given in [3]. We say that a code C is of even weight if and only if all of its codewords are of even weight.

Definition 2. *Given binary codes C_1 and C_2 of length n with $C_2 \subseteq C_1$, $Q(C_1,C_2)$ is a CSS-T code if and only if C_2 is of even weight and for all codewords $c \in C_2$, the shortening of $C_1^\perp$ with respect to the coordinates of $[n] \setminus supp(c)$ contains a self-dual code.*

We use the following result from [6] to determine quantum codes which are CSS-T.

Theorem 3. *Let C_1 and C_2 be binary codes of length n. The following are equivalent.*

1. *$Q(C_1,C_2)$ is a CSS-T code.*
2. *$C_2 \subseteq C_1 \cap (C_1^2)^\perp$.*
3. *$C_1^\perp + C_1^2 \subseteq C_2^\perp$.*

Moreover, if $Q(C_1,C_2)$ is a CSS-T code, then C_2 is self-orthogonal.

We use the following result from [6] to determine the parameters of a quantum CSS-T code.

Corollary 4. *Let (C_1,C_2) be a CSS-T pair. Then*

$$\min\{\mathrm{wt}(C_1), \mathrm{wt}(C_2^\perp)\} = \mathrm{wt}(C_2^\perp),$$

and the parameters of the corresponding CSS-T code are

$$[[n, k_1-k_2, \geq \mathrm{wt}(C_2^\perp)]].$$

Moreover, if the code is nondegenerate, we have equality in the minimum distance.

Definition 5. *A pair of codes (C_1,C_2) satisfying the conditions in Theorem 3 is called a CSS-T pair.*

According to Theorem 3, given a code C, the intersection $C \cap (C^2)^\perp$ is useful in determining a maximal CSS-T pair (C_1,C_2) with $C_1 = C$. To better understand this intersection, we note its relationship to the hull of a code.

Definition 6. *The hull of a code C is* $\mathrm{Hull}(C) := C \cap C^\perp$. *Given codes C_1 and C_2 of the same length, the relative hull of C_1 with respect to C_2 is* $\mathrm{Hull}_{C_2}(C_1) := C_1 \cap C_2^\perp$.

Lemma 7. *For a binary code C,*

$$C \cap (C^2)^\perp = \mathrm{Hull}(C) \cap \mathrm{Hull}(C^2).$$

Proof. Note that $C \subseteq C^2$, since $w \star w = w$ for all $w \in \mathbb{F}_2^n$. Note $C \cap (C^2)^\perp \subseteq C^2 \cap (C^2)^\perp = \mathrm{Hull}(C^2)$. In addition, $(C^2)^\perp \subseteq C^\perp$ and $C \cap (C^2)^\perp \subseteq \mathrm{Hull}(C)$. Thus, $C \cap (C^2)^\perp \subseteq \mathrm{Hull}(C) \cap \mathrm{Hull}(C^2)$.

As $\mathrm{Hull}(C) \cap \mathrm{Hull}(C^2) = C \cap C^\perp \cap C^2 \cap (C^2)^\perp \subseteq C \cap (C^2)^\perp$, the proof is complete. □

We come to one of the main results of this section. Lemma 7 provides another characterization of CSS-T codes, as stated in the next result.

Theorem 8. *Given binary codes C_1 and C_2 with $C_2 \subseteq C_1$, $Q(C_1, C_2)$ is a CSS-T code if and only if*

$$C_2 \subseteq \mathrm{Hull}(C_1) \cap \mathrm{Hull}(C_1^2)$$

if and only if

$$C_2 \subseteq \mathrm{Hull}_{C_1^2}(C_1).$$

Proof. This result follows immediately from Theorem 3, Lemma 7, and Definition 6. □

Notice that Theorem 8 together with Theorem 3 indicates that to determine a CSS-T pair (C_1, C_2), one may restrict the search to self-orthogonal codes C_2 in the relative hull of C_1 with respect to its square.

IV. Puncturing

Let $C \subset \mathbb{F}_2^n$ be a code and $i \in [n]$. The puncturing of C in $\{i\}$, denoted by $C|_i$, is the binary code

$$\begin{aligned} C|_i := \{&(c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n) \\ &: (c_1, \ldots, c_{i-1}, c_i, c_{i+1}, \ldots, c_n) \in C, \\ &\text{for some } c_i \in \mathbb{F}_2\}. \end{aligned}$$

For $S \subset [n]$, we write $C|_S$ for the successive puncturing of C in the coordinates indexed by the elements in S. The code C is degenerated on $\{i\}$ if $c_i = 0$ for every $c = (c_1, \ldots, c_i, \ldots, c_n) \in C$.

We come to one of the main results of this section. The following theorem states that if $Q(C_1, C_2)$ is a CSS-T code, then $Q(C_1|_i, C_2|_i)$ is a CSS-T code whenever the code $\mathrm{Hull}(C_1) \cap \mathrm{Hull}(C_1^2)$ is degenerated on $\{i\}$.

Theorem 9. *Let $C_2 \subseteq C_1$ be binary codes. Assume that C_2 is degenerated on $\{i\}$. If $Q(C_1, C_2)$ is a CSS-T code, then $Q(C_1|_i, C_2|_i)$ is a CSS-T code.*

Proof. Note that for any $c \in C_2 \subseteq C_1$, $c_i = 0$, so $c|_i \in C_2|_i$ and $c|_i \in C_1|_i$.

As $Q(C_1, C_2)$ is a CSS-T code, then

$$C_2 \subseteq \mathrm{Hull}(C_1) \cap \mathrm{Hull}(C_1^2)$$

by Theorem 8. Thus, $c \cdot w = c|_i \cdot w|_i = 0$ for every element w in C_1^2. We obtain

$$C_2|_i \subseteq C_1|_i \cap (C_1|_i)^{2\perp},$$

from which we get the conclusion by Theorem 8. □

The support of $C \subset \mathbb{F}_2^n$ is denoted and defined by

$$\begin{aligned} \mathrm{Supp}(C) := \{&i \in [n] : c_i \neq 0 \text{ for some} \\ &c = (c_1, \ldots, c_i, \ldots, c_n) \in C\}. \end{aligned}$$

Corollary 10. *Let S be the complement of $\mathrm{Supp}(C)$. If $Q(C_1, C_2)$ is a CSS-T code, then $(C_1|_S, C_2|_S)$ is a CSS-T code.*

Proof. This is a consequence of Theorem 9. □

V. Quasi-cyclic low-density codes

In this section, we study quantum LDPC and LDGM codes defined by quasi-cyclic codes.

Definition 11. *A CSS code $Q(C_1, C_2)$ is a quantum LDPC code if C_1 and $C_2^\perp$ are LDPC codes, or, equivalently, if C_1 is an LDPC code and C_2 is an LDGM code. Similarly, $Q(C_1, C_2)$ is a quantum LDGM code if C_1 and $C_2^\perp$ are LDGM codes, or, equivalently, if C_1 is an LDGM code and C_2 is an LDPC code.*

Remark 12. *Note that a binary generator matrix for the stabilizer that defines a quantum LDPC code $Q(C_1, C_2)$ can be written as*

$$\begin{bmatrix} 0 & H_1 \\ G_2 & 0 \end{bmatrix},$$

where H_1 and G_2 are sparse matrices.

Here we use those ideas to consider LDPC codes which give rise to CSS-T pairs. As proof of concept, we make use of a code construction found in [22], where the authors sought codes that have both efficient encoding algorithms and fast iterative decoding algorithms.

For an integer $L \geq 2$, define the matrix

$$P := \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{F}_2^{L \times L}.$$

Take positive integers m and n. Let $a_{ij} \in \mathbb{Z}_L \cup \{\infty\}$, for $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$. Define a code C by the parity check matrix

$$H = \begin{pmatrix} P^{a_{00}} & P^{a_{01}} & \cdots & P^{a_{0(n-1)}} \\ \vdots & \vdots & \ddots & \vdots \\ P^{a_{(m-1)0}} & P^{a_{(m-1)1}} & \cdots & P^{a_{(m-1)(n-1)}} \end{pmatrix}, \quad (2)$$

where P^∞ denotes a square matrix of zeroes of size L and $P^{a_{ij}}$ is the usual a_{ij} power matrix multiplication of the matrix P.

Observe that if $a \in \mathbb{Z}_L$, then P^a is a permutation matrix. Indeed, $\mathrm{Row}_i\ P^a = e_{a+i \bmod L}^T$. Note that the code C is a quasi-cyclic LDPC (QC-LDPC) code. The weight of each row is $\leq n$ and the weight of each column is $\leq m$. Moreover, by

knowing a, we can recover immediately P^a. Thus, we can store H with a smaller base matrix

$$M_H = \begin{pmatrix} a_{00} & \cdots & a_{0(n-1)} \\ \vdots & \ddots & \vdots \\ a_{(m-1)0} & \cdots & a_{(m-1)(n-1)} \end{pmatrix} \in (\mathbb{Z}_L \cup \{\infty\})^{m\times n}.$$

We aim to describe the square of C in terms of the entries of M_H. To that end, for $a \in (\mathbb{Z}_L \cup \{\infty\})^n$, let $R(a) \in \mathbb{F}_2^{Ln}$ be defined as

$$[R(a)]_i = \begin{cases} 1 & \text{if } i = jL + a_j \text{ for some } j \in \{0, \cdots, n-1\} \\ & \text{such that } a_j \neq \infty \\ 0 & \text{otherwise.} \end{cases}$$

Making use of this definition, the following Lemma gives a natural connection between M_H and the rows of H.

Lemma 13. *Let C be a quasi-cyclic LDPC code with shift L and assume $a \in (\mathbb{Z}_L \cup \{\infty\})^n$ is such that $R(a) \in C$. Extend the sum over $\mathbb{Z}_L$ to $\mathbb{Z}_L \cup \{\infty\}$ by taking $x + \infty = \infty$ for any $x \in \mathbb{Z}_L \cup \{\infty\}$ and take $\mathbf{1} = (1, \ldots, 1) \in \mathbb{Z}_L^n$. Then*

$$R(a + j\mathbf{1}) \in C, \ \forall j \in \mathbb{Z}_L.$$

Proof. Observe that

$$R(a_i + j) = \begin{cases} R(a_i + j) & \text{if } a_i \neq \infty \\ R(\infty) & \text{if } a_i = \infty. \end{cases}$$

In either case, $R(a_i + j) = R(a_i) \cdot P$ and thus $R(a + j\mathbf{1}) \in C$. □

Consider the operation $* : (\mathbb{Z}_L \cup \{\infty\})^2 \to \mathbb{Z}_L \cup \{\infty\}$ defined by

$$a * b = \begin{cases} a & \text{if } a = b \\ \infty & \text{otherwise.} \end{cases}$$

The operation extends naturally component-wise to $(\mathbb{Z}_L \cup \{\infty\})^n$. Moreover, there is a relationship between $\star$ and $*$, as shown in the next result.

Proposition 14. *Given $a, b \in (\mathbb{Z}_L \cup \{\infty\})^n$,*

$$R(a) \star R(b) = R(a * b).$$

Proof. Let $a = (a_0, \ldots, a_{n-1})$, $b = (b_0, \ldots, b_{n-1})$ and $0 \leq i \leq n-2$. We will focus on the entries of $R(a)$ indexed by $iL + (i+1)L - 1$, meaning $R(a_i)$. Observe that $R(a_i) = e_{a_i}$, the a_i^{th} standard basis vector in $\mathbb{F}_2^L$. Thus,

$$R(a_i) \star R(b_i) = e_{a_i} \star e_{b_i} = \begin{cases} 0 & \text{if } a_i \neq b_i \\ e_{a_i} & \text{otherwise.} \end{cases}$$

Thus,

$$R(a_i) \star R(b_i) = R(a_i * b_i).$$

Since $R(a)$ is the concatenation of $R(a_i)$, we have the conclusion. □

Proposition 15. *Given the code $C^\perp$ defined by parity check matrix H as in Equation (2) and its corresponding matrix M_H with rows $A_1, \ldots, A_m$, its square $(C^\perp)^2$ is generated by a matrix $H' \in \mathbb{F}_2^{mL\times nL}$ such that $M_{H'} \in (\mathbb{Z}_L \cup \{\infty\})^{m\times n}$ has rows $A_i * (A_j + h\mathbf{1})$ for any $0 \leq h \leq L-1$.*

Proof. We know that $C^\perp$ is generated by $R(A_i + j\mathbf{1})$ with $0 \leq j \leq L-1$. Thus, by Proposition 14, the square is generated by

$$R(A_{i_1} + j_1\mathbf{1}) \star R(A_{i_2} + j_2\mathbf{1}) = R((A_{i_1} + j_1\mathbf{1}) * (A_{i_2} + j_2\mathbf{1})).$$

The conclusion will follow if $w = (A_{i_1} + j_1\mathbf{1}) * (A_{i_2} + j_2\mathbf{1}) = A_{i_1} * (A_{i_2} + h\mathbf{1}) + h'\mathbf{1}$ for some $h, h' \in \mathbf{Z}_L$. We claim that

$$w = A_{i_1} * (A_{i_2} + (j_2 - j_1)\mathbf{1}) + j_1\mathbf{1}.$$

Observe that if $(A_{i_1})_\nu + j_1 = (A_{i_2})_\nu + j_2$ then either $(A_{i_1})_\nu = (A_{i_2})_\nu = \infty$ or $(A_{i_1})_\nu = (A_{i_2})_\nu + (j_2 - j_1)$. In the first case, $w_\nu = \infty$ and in the second case $w_\nu = (A_{i_2})_\nu + j_2$. In any case, we have

$$w_\nu = (A_{i_1})_\nu * ((A_{i_2})_\nu + (j_2 - j_1)) + j_1.$$

On the other hand, if $(A_{i_1})_\nu + j_1 \neq (A_{i_2})_\nu + j_2$, then necessarily $w_\nu = \infty$. If $(A_{i_1})_\nu \neq \infty$, then $(A_{i_1})_\nu \neq (A_{i_2})_\nu + (j_2 - j_1)$ and

$$w_\nu = (A_{i_1})_\nu * ((A_{i_2})_\nu + (j_2 - j_1)) + j_1.$$

Similarly for $(A_{i_2})_\nu \neq \infty$ and then we have the conclusion.

Since $R(w) \in (C^\perp)^{*2}$, by Lemma 13, we have $R(w + h\mathbf{1}) \in (C^\perp)^2$ for any h and thus, we can store any of them to build $M_{H'}$, from where we have the conclusion by taking $h = -j_1 \bmod L$. □

VI. Code

In this section, we present Magma [23] code based on Theorem 9 and quasi-cyclic codes to find quantum LDPC or LDGM CSS-T codes computationally. We also describe the algorithms in case one wishes to use a different software, for instance Macaulay2 [24] along with the coding theory package [25].

We start by giving the algorithms to compute the Schur product between two matrices, the square (respect the Schur product) of a matrix, and the H matrix given in Eq. 2.

```
function POINTWISE(A, B)        ▷ Returns the pointwise matrix between same-size matrices A and B
    n ← |{columns of A}|
    m ← |{rows of A}|
    C ← 0_{n×m}
    for j ∈ [n], i ∈ [m] do
        C[i, j] ← A[i, j] * B[i, j]
    end for
    return C
end function
```

We now present the Magma code that can be used to find LDPC or LDGM quantum CSS-T codes. The following function returns the Schur product between same-size matrices A and B.

function SQUARE(A) ▷ Returns the (Schur) square of A
    $n \leftarrow |\{\text{columns of } A\}|$
    $m \leftarrow |\{\text{rows of } A\}|$
    $C \leftarrow 0_{m^2 \times n}$
    $\ell \leftarrow 1$
    for $i, j \in [m]$ **do**
        $C[\ell] \leftarrow$ Pointwise(Row i of A, Row j of A)
        $\ell \leftarrow \ell + 1$
    end for
    return C
end function

```
function Pointwise(A,B)
 return Matrix([[A[i,j]*B[i,j] :
 j in [1..Ncols(A)]] :
 i in [1..Nrows(A)]]);
end function;
```

The following function returns the square of the matrix A (in terms of the Schur product).

```
function Square(A)
 return Matrix([Pointwise(RowSubmatrix
 (A, i, 1),RowSubmatrix(A, j, 1))
 : i,j in [1..Nrows(A)]]);
end function;
```

The following function returns the quasi-cyclic low-density matrix H defined by the integer L and the matrix A (see Eq. 2) that can be used as generator matrix for a QC-LDGM code or parity-check matrix for a QC-LDPC code.

```
function QCLD(L,A)
 P:=ZeroMatrix(FiniteField(2), L, L);
 InsertBlock(~P, IdentityMatrix
 (FiniteField(2), L-1), 1, 2);
 P[L,1]:=FiniteField(2)!1;

 H:=ZeroMatrix(FiniteField(2),
 Nrows(A)*L, Ncols(A)*L);
 for i in [0..Nrows(A)-1] do
  for j in [0..Ncols(A)-1] do
  InsertBlock(~H,P^A[i+1,j+1],
  i*L+1, j*L+1);
  end for;
 end for;
 return H;
end function;
```

Example 16. *We will use the previous functions to generate a CSS-T code. Specifically, we use the QCLD function to provide a sparse generator matrix.*

```
L:=4;
A:= Matrix(IntegerRing(),
2, 4, [3,1,2,1,  3,3,2,3] );

G:=QCLD(L,A);
C1:=LinearCode(G);
C2:=C1 meet Dual(LinearCode
(Square(GeneratorMatrix(C1))));
C1;
Dual(C2);
```

Then C_1 is a $[16, 6, 4]$ binary code with generator matrix $G_1 =$

$$\begin{bmatrix} 1&0&0&0&0&0&1&0&0&0&0&1&0&0&1&0\\ 0&1&0&0&0&0&0&1&1&0&0&0&0&0&0&1\\ 0&0&1&0&0&0&1&0&0&1&0&0&0&0&1&0\\ 0&0&0&1&0&0&0&1&0&0&1&0&0&0&0&1\\ 0&0&0&0&1&0&1&0&0&0&0&0&1&0&1&0\\ 0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&1 \end{bmatrix}.$$

Moreover,

$$C_2 = \mathrm{Hull}(C_1) \cap \mathrm{Hull}(C_1^2) = C_1$$

and $C_2^\perp$ is a $[16, 10, 2]$ binary code. So, by Corollary 4 and Theorem 8, the quantum code $Q(C_1, C_2)$ is a $[[16, 0, \geq 2]]$ CSS-T code.

As an additional example of the techniques introduced in this paper, we provide the following.

Example 17. Let C_1 and C_2 be defined by the generator matrices

$$\begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\ 1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0\\ 1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0\\ 1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0 \end{bmatrix}$$

and

$$\begin{bmatrix} 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\ 1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0\\ 1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0\\ 1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0 \end{bmatrix},$$

respectively. We can check that (C_1, C_2) is a CSS-T using Theorem 8 and [23]. As the code C_2 is degenerated with respect to the last column, we can puncture that column to obtain that $((C_1)|_{15}, (C_2)_{15})$ is also a CSS-T pair by Theorem 9. The minimum distance of $((C_2)|_{15})^\perp$ is 3 and then we get a CSS-T code with parameters $[[15, 1, 3]]$.

VII. CONCLUSION

In this paper, we provided a characterization of CSS-T codes using the relative hull of a code with respect to its square. We proved that under certain conditions, we can puncture the component codes of a CSS-T pair to obtain another CSS-T pair. We considered the use of quasi-cyclic codes to design LDPC and LDGM quantum CSS-T codes computationally. Toy examples were given as a proof of concept, demonstrating a possible step towards obtaining CSS-T codes using the characterization.

REFERENCES

[1] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, pp. 1098–1105, Aug 1996. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.54.1098

[2] A. M. Steane, "Error correcting codes in quantum theory," *Physical Review Letters*, vol. 77, no. 5, p. 793, 1996.

[3] N. Rengaswamy, R. Calderbank, M. Newman, and H. D. Pfister, "Classical coding problem from transversal T gates," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 1891–1896.

[4] E. Andrade, J. Bolkema, T. Dexter, H. Eggers, V. Luongo, F. Manganiello, and L. Szramowski, "CSS-T codes from Reed Muller codes for quantum fault tolerance," *ArXiv 2305.06423*, 2023.

[5] E. Berardini, A. Caminata, and A. Ravagnani, "Structure of CSS and CSS-T quantum codes," *ArXiv 2310.16504*, 2023.

[6] E. Camps-Moreno, H. H. López, G. L. Matthews, D. Ruano, R. San-José, and I. Soprunov, "The poset of binary CSS-T quantum codes and cyclic codes," 2023.

[7] N. P. Breuckmann and J. N. Eberhardt, "Quantum low-density parity-check codes," *PRX Quantum*, vol. 2, p. 040101, Oct 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PRXQuantum.2.040101

[8] D. Gottesman, "Fault-tolerant quantum computation with constant overhead," *Quantum Info. Comput.*, vol. 14, no. 15–16, p. 1338–1372, nov 2014.

[9] Q. Xu, J. P. B. Ataides, C. A. Pattison, N. Raveendran, D. Bluvstein, J. Wurtz, B. Vasic, M. D. Lukin, L. Jiang, and H. Zhou, "Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays," 2023.

[10] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, "High-threshold and low-overhead fault-tolerant quantum memory," 2024.

[11] S. B. Bravyi and A. Y. Kitaev, "Quantum codes on a lattice with boundary," 1998.

[12] J.-P. Tillich and G. Zémor, "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1193–1202, 2014.

[13] P. Panteleev and G. Kalachev, "Asymptotically good quantum and locally testable classical LDPC codes," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 375–388. [Online]. Available: https://doi.org/10.1145/3519935.3520017

[14] A. Leverrier and G. Zemor, "Quantum tanner codes," in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. Los Alamitos, CA, USA: IEEE Computer Society, nov 2022, pp. 872–883. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/FOCS54457.2022.00117

[15] S. Evra, T. Kaufman, and G. Zémor, "Decodable quantum LDPC codes beyond the $\sqrt{n}$ distance barrier using high-dimensional expanders," *SIAM Journal on Computing*, vol. 0, no. 0, pp. FOCS20–276–FOCS20–316, 0. [Online]. Available: https://doi.org/10.1137/20M1383689

[16] A. Krishna, I. L. Navon, and M. Wootters, "Viderman's algorithm for quantum LDPC codes," 2023.

[17] G. Luo, X. Cao, and X. Chen, "MDS codes with hulls of arbitrary dimensions and their quantum error correction," *IEEE Transactions on Information Theory*, vol. 65, no. 5, pp. 2944–2952, 2019.

[18] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. L. M. Barreto, "MDPC-McEliece: New McEliece variants from moderate density parity-check codes," in *2013 IEEE International Symposium on Information Theory*, 2013, pp. 2069–2073.

[19] E. Assmus and J. Key, "Affine and projective planes," *Discrete Mathematics*, vol. 83, no. 2, pp. 161–187, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0012365X9090003Z

[20] J. L. Massey, "Linear codes with complementary duals," *Discrete Mathematics*, vol. 106-107, pp. 337–342, 1992. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0012365X9290563U

[21] S. E. Anderson, E. Camps-Moreno, H. H. López, G. L. Matthews, D. Ruano, and I. Soprunov, "Relative hulls and quantum codes," 2023.

[22] S. Myung, K. Yang, and J. Kim, "Quasi-cyclic LDPC codes for fast encoding," *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2894–2901, 2005.

[23] W. Bosma, J. Cannon, and C. Playoust, "The Magma algebra system. I. The user language," *J. Symbolic Comput.*, vol. 24, no. 3-4, pp. 235–265, 1997. [Online]. Available: http://dx.doi.org/10.1006/jsco.1996.0125

[24] D. R. Grayson and M. E. Stillman, "Macaulay2, a software system for research in algebraic geometry." [Online]. Available: http://www.math.uiuc.edu/Macaulay2/

[25] T. Ball, E. Camps, H. Chimal-Dzul, D. Jaramillo-Velez, H. H. López, N. Nichols, M. Perkins, I. Soprunov, G. Vera-Martínez, and G. Whieldon, "Coding theory package for Macaulay2," *Journal of Software for Algebra and Geometry*, to appear.