CLASSES OF HIGH-PERFORMANCE QUANTUM LDPC CODES
FROM FINITE PROJECTIVE GEOMETRIES

by

Jacob M. Farinholt
A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Master of Science
Mathematics

Committee:

Dr. Geir Agnarsson, Thesis Director

Dr. Walter Morris, Committee Member

Dr. Marco Lanzagorta, Committee Member

Dr. Stephen Saperstone, Chairman,
Department of Mathematical Sciences

Dr. Timothy L. Born, Associate Dean for
Student and Academic Affairs, College
of Science

Dr. Vikas Chandhoke, Dean, College of
Science

Date: July 5, 2012

Summer Semester 2012
George Mason University
Fairfax, VA

Classes of High-Performance Quantum LDPC Codes From Finte Projective Geometries

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science at George Mason University

By

Jacob M. Farinholt
Bachelor of Science
University of Mary Washington, 2009

Director: Geir Agnarsson, Associate Professor
Department of Mathematical Sciences

Summer Semester 2012
George Mason University
Fairfax, VA

This research was funded by Naval Surface Warfare Center, Dahlgren Division (NSWCDD) In-house Laboratory Independent Research (ILIR) program, and NSWCDD's Academic Fellowship Program. This thesis is also available as an NSWCDD Technical Report, NSWCDD-TR-12-00132.

Dedication

I dedicate this thesis to my wife, Sarah.

Acknowledgments

I would like to thank all the members of my thesis committee for their thoughtful reviews; Geir Agnarsson for his patience, support, and willingness to jump into a new subject area; Keith Mellinger and Keye Martin for their thoughtful reviews and suggestions; James Troupe for teaching me almost everything I know about quantum information theory, and for spending countless hours listening to me talk about my research, while providing helpful insights and direction. Finally, I would like to thank my wife for her unending patience and support, without which I could have never completed this.

The author acknowledges funding support from Naval Surface Warfare Center (NSWCDD) Academic Fellowship Program (AFP) and In-house Laboratory Independent Research (ILIR) program.

Table of Contents

List of Tables

List of Figures

Abstract

CLASSES OF HIGH-PERFORMANCE QUANTUM LDPC CODES FROM FINTE PROJECTIVE GEOMETRIES

Jacob M. Farinholt, M.S.

George Mason University, 2012

Thesis Director: Dr. Geir Agnarsson

Due to their fast decoding algorithms, quantum generalizations of low-density parity check, or LDPC, codes have been invesitgated as a solution to the problem of decoherence in fragile quantum states [1,2]. However, the additional twisted inner product requirements of quantum stabilizer codes force four-cycles and eliminate the possibility of randomly generated quantum LDPC codes. Moreover, the classes of quantum LDPC codes discovered thus far generally have unknown or small minimum distance, or a fixed rate (see [3, 4] and references therin). This paper presents several new classes of quantum LDPC codes constructed from finite projective planes. These codes have rates that increase with the block length n and minimum weights proportional to $n^{1/2}$. For the sake of completeness, we include an introduction to classical error correction and LDPC codes, and provide a review of quantum communication, quantum stabilizer codes, and finite projective geometry.

Chapter 1: Classical Error Correction

We begin with a brief overview of classical error correcting codes. If a more detailed introduction is desired, we refer the reader to [5]. In classical error correction, information is made up of sequences of bits. We generally assume that noise (i.e. bit flips) acts independently on each bit, and the probability of a bit flipping is given by $p \in [0, 1]$. It follows that the probability of a set of k bits flipping in a transmission is given by p^k. While not every error can be corrected, as k increases, the probability that k bits get flipped decreases rapidly. For large enough k, we can assume that the probability that more than k bits flipping is so small that we are willing to take that chance.

One may naively try to correct errors on bits by simply using the repetition code, i.e. repeating each bit a fixed number of times,

$$1 \mapsto 11\ldots1, \qquad 0 \mapsto 00\ldots0,$$

and then doing majority vote to determine which bit was actually sent. While this generally works, it is in fact extremely inefficient.

1.1 The Encoding Process

Instead, assume we have a k-dimensional *binary vector space* V_k that corresponds to our dictionary, that is, a k-dimensional vector space with binary inputs, and addition defined point-wise modulo 2. We then define an injective linear map $\varphi : V_k \hookrightarrow V_n$ into a larger binary vector space of dimension $n > k$. We do this in the hopes that the extra vectors in V_n that are not in the image of V_k correspond to identifiable errors. The image $\varphi(V_k)$ is called the *code space* of V_n.

We define the *Hamming distance* between two vectors (or *distance* for short) to be the number of bits in which they differ, i.e. the weight of the sum of the two vectors. For example, consider the two vectors $v_1 = 0010$ and $v_2 = 1011$. Then $d(v_1, v_2) = wt(v_1 + v_2) = wt(1001) = 2$. The *minimum distance*, d, of a code is defined to be the smallest distance between any two distinct codewords. A code $\mathcal{C} = \varphi(V_k) \subseteq V_n$ having minimum distance d is called an $[n, k, d]$ code. Observe that, since the code is closed under addition, the minimum distance is the same as the weight of the smallest weight nonzero codeword. Thus the terms "minimum weight" and "minimum distance" can be used interchangeably.

In general, we assume that in a given transmission the fewest number of errors always occurred. In other words, if an error vector is received, we assume the correct codeword is the one "closest" to it. Given this, how many bit-flip errors can be detected and corrected in an $[n, k, d]$ code $\mathcal{C}$? Clearly, up to $d - 1$ bit flip errors can always be detected. Suppose v_1 and v_2 are two codewords which are distance d apart from each other. The largest non-intersecting spheres around v_1 and v_2 have radius $t = \lfloor \frac{d-1}{2} \rfloor$. Hence, if an error vector inside the sphere around v_1 is received, we assume the correct codeword is v_1. We conclude, then, that if the code has minimum distance d, it can always correct up to $t = \lfloor \frac{d-1}{2} \rfloor$ and fewer errors.

The general goal of coding theory is to try to make $n - k$ as small as possible (i.e. add as few non-information bits as possible), while making d as large as possible; however, we are limited by the following bound, known as the *Singleton bound*:

$$n - k \geq d - 1.$$

This bound is well-known, and the proof is straightforward [5]. Codes that saturate this bound are known as *MDS* (maximum distance separable) codes.

1.2 Checking for Errors: Syndrome Decoding

A *parity check matrix* H for an $[n, k, d]$ code $\mathcal{C} \subseteq V_n$ is a matrix whose rows form a collection of vectors in V_n that span the orthogonal space (or dual space) of $\mathcal{C}$, denoted $\mathcal{C}^{\perp}$. E.g. $v \in \mathcal{C}$ if and only if $vH^T = \overline{0}$. It follows from elementary linear algebra, then, that if $\mathcal{C}$ has dimension k, then $\dim(\mathcal{C}^{\perp}) = n - k$. By this construction, $\mathcal{C} = \ker(H)$, the kernel of H, as H acts on V_n. If an error vector $e \notin \mathcal{C}$ is received, then at least one row of H will not be orthogonal to e, so that $eH^T \neq \overline{0}$.

The vector $m = vH^T$ is called the *syndrome* of the vector $v \in V_n$. The syndromes separate all of V_n into distinct cosets, which correspond to the locations of the errors.

1.3 Example: Hamming $[7, 4, 3]$ Code

The Hamming $[7, 4, 3]$ code has a parity check matrix given by:

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{1.1}$$

This matrix separates the binary vector space V_7 into 8 distinct cosets. We explicitly list the cosets, with corresponding syndrome values, in Table 1.1. The first element of each row in Table 1.1 is a codeword, while the remaining elements in the row correspond to the particular errors that are corrected to that codeword. By inspection, we see that every coset corresponds to the location of a particular single bit-flip. As one should expect, if more than one bit-flip error occurs, then this code cannot correct it.

Table 1.1: The syndrome cosets of the Hamming [7,4,3] code. The first coset is the actual code. Each row contains all of the elements of V_7 that will be corrected to the first entry of the row.

000	011	101	110	111	100	010	001
0000000	1000000	0100000	0010000	0001000	0000100	0000010	0000001
1000011	0000011	1100011	1010011	1001011	1000111	1000001	1000010
0100101	1100101	0000101	0110101	0101101	0100001	0100111	0100100
0010110	1010110	0110110	0000110	0011110	0010010	0010100	0010111
0001111	1001111	0101111	0011111	0000111	0001011	0001101	0001110
1100110	0100110	1000110	1110110	1101110	1100010	1100100	1100111
1010101	0010101	0110101	1000101	1011101	1010001	1010111	1010100
1001100	0001100	1101100	1011100	1000100	1001000	1001110	1001101
0110011	1110011	0010011	0100011	0111011	0110111	0110001	0110010
0101010	1101010	0001010	0111010	0100010	0101110	0101000	0101011
0011001	1011001	0111001	0001001	0010001	0011101	0011011	0011000
1110000	0110000	1010000	1100000	1111000	1110100	1110010	1110001
1101001	0101001	1001001	1111001	1100001	1101101	1101011	1101000
1011010	0011010	1111010	1001010	1010010	1011110	1011000	1011011
0111100	1111100	0011100	0101100	0110100	0111000	0111110	0111101
1111111	0111111	1011111	1101111	1110111	1111011	1111101	1111110

1.4 Classical Low-Density Parity Check Codes

Classical *low-density parity check* (LDPC) codes were first discovered by Gallager [6] in 1960. Later, it was shown that describing these codes with bipartite graphs (called *Tanner graphs* in the literature) made it easier to understand how well the codes work under their iterative belief propagation decoding algorithm. These classical error correcting codes are currently some of the best known, having rates asymptotically approaching the Shannon limit [7]. We say that a code is a low-density parity check code, or LDPC code, if its parity check matrix, H, is sparse[1], and any two rows of H have very few (preferably no more than one) "ones" in common positions.

An LDPC code can be analyzed by constructing a bipartite graph in which the left

[1]While there is no strict definition, a binary matrix is generally considered to be sparse when the ratio of "ones" to "zeros" is relatively small, at least less than 1/2.

vertices (called "bit nodes") correspond to columns of the parity check matrix and the right vertices (called "check nodes") correspond to the rows of the parity check matrix. A line connecting check node i with bit node j in the Tanner graph, corresponds to a 1 in position i, j of the parity check matrix. It follows that if the codewords of an LDPC code have length n, then there will be n bit nodes in the graph, and each check node corresponds to a particular row of the parity check matrix.

While Tanner graphs are helpful in analyzing decoding algorithms for LDPC codes, they can also be used to define codewords. Suppose each entry x_i of a length n binary vector $v = (x_1, x_2, \ldots, x_n)$ is placed in the corresponding bit node i in a Tanner graph with n bit nodes. For each check node, consider the binary sum of the entries in each bit node connected to it. If these sums are zero at each check node, then the vector v is a codeword for the corresponding LDPC code. For example, if 011110 is put into the bit nodes in Figure 1.1, we can see that the sums at each check node are zero, and hence this vector is a codeword.

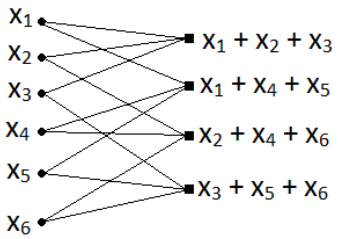

Figure 1.1: A Tanner graph of an LDPC code

Given the Tanner graph of Figure 1.1, we can construct the corresponding parity check

using the method described above:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \tag{1.2}$$

Note that the check node sums provide a syndrome, and the sums at each check node are all zero precisely when the vector is in the kernel of the parity check matrix.

If two rows of the parity check matrix have multiple "ones" in common position, then the corresponding Tanner graph will have a four-cycle. Four-cycles in Tanner graphs generally cause poor performance in the iterative decoding algorithms used with LDPC codes. For this reason, one of the primary goals when constructing LDPC codes is to minimize or eliminate four-cycles in the Tanner graph.

Chapter 2: Review of Quantum Information Theory

We now want to consider codes in the quantum realm. Although an effort is made to be clear and concise here, we refer the reader to [8] if a slightly more detailed overview is desired. In quantum communication, as we will see, errors are much more arbitrary than those that can occur in a classical information channel. In order to understand errors in a quantum information channel, we will first characterize states of a quantum system.

A *quantum system* is a quantum particle or collection of particles that have some measurable property, which we call the *state* of the system, represented by $|\psi\rangle$. A single particle pure two-state system is called a *qubit*. Its state is represented as a convex linear combination of basis elements $|b_i\rangle$ in the complex Hilbert space $\mathcal{H}_2$, so $|\psi\rangle = \alpha_0|b_0\rangle + \alpha_1|b_1\rangle$, where $|b_0\rangle$ and $|b_1\rangle$ form a basis for $\mathcal{H}_2$, α_0 and α_1 are in $\mathbb{C}$, and $|\alpha_0|^2 + |\alpha_1|^2 = 1$. One important fact about quantum states is that there is absolutely no distinction made between $|\psi\rangle$ and $e^{i\theta}|\psi\rangle$ for any real θ. The term $e^{i\theta}$ is called a *global phase*.

We can generalize the definition of a single qubit state to an n-qubit state by defining $|\psi\rangle$ to be a linear combination of basis elements $|b_i\rangle$ in $\mathcal{H}_2^n$ (where $\mathcal{H}_2^n = \mathcal{H}_2^{\otimes n} = \mathcal{H}_2 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_2$, a tensor product over $\mathbb{C}$), $|\psi\rangle = \alpha_0|b_0\rangle + \alpha_1|b_1\rangle + \cdots + \alpha_{n-1}|b_{n-1}\rangle$ where $|\alpha_0|^2 + \cdots + |\alpha_{n-1}|^2 = 1$. The basis elements are then simply tensor products of basis elements in $\mathcal{H}_2$. In other words, if we denote a pair of basis elements in $\mathcal{H}_2$ by $|0\rangle$ and $|1\rangle$ so that a qubit looks like $\alpha_0|0\rangle + \alpha_1|1\rangle$, then a 2-qubit system can be represented in terms of the basis elements $|0\rangle \otimes |0\rangle = |00\rangle$, $|0\rangle \otimes |1\rangle = |01\rangle$, $|1\rangle \otimes |0\rangle = |10\rangle$, and $|1\rangle \otimes |1\rangle = |11\rangle$, and similarly for larger systems.

It is also important to note that, although any single qubit can be written in terms of a particular basis, it is sometimes convenient to interpret a state in terms of a different basis.

Two bases of particular importance in quantum information theory are given by

$$\text{basis 1: } |0\rangle_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and

$$\text{basis 2: } |0\rangle_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{|0\rangle_1 + |1\rangle_1}{\sqrt{2}},$$

$$|1\rangle_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{|0\rangle_1 - |1\rangle_1}{\sqrt{2}}.$$

The $\frac{1}{\sqrt{2}}$ term in basis 2 is the normalization constant. Suppose a state $|\psi\rangle$ is prepared in basis 2, $|\psi\rangle = \alpha|0\rangle_2 + \beta|1\rangle_2$. Then in basis 1 the state becomes

$$|\psi\rangle = \frac{\alpha+\beta}{\sqrt{2}}|0\rangle_1 + \frac{\alpha-\beta}{\sqrt{2}}|1\rangle_1.$$

These bases have a very physical meaning. In a certain aspect, the state of a system represents one's knowledge of the system. For example, two measurable properties of a particular quantum system may be position and momentum. Each of these measurable properties has a representation in a particular basis. We know from the Heisenberg uncertainty principle, that the more precisely we determine a particle's position, the less preciscely we can determine any information about its momentum, and conversely. If basis 1 corresponds to a particle's position and basis 2 corresponds to the particle's momentum, then suppose that the particle was prepared in the state $|0\rangle_2$. Suppose that no errors occurred to the state before it was measured. If it was measured with respect to basis 2, then we would obtain exactly the state $|0\rangle_2$, implying that we know everything about the

particle's momentum. What would have happened if we had instead measured the state with respect to basis 1? We would obtain the state $|0\rangle_1$ with probability $|1/\sqrt{2}|^2 = 1/2$ and state $|1\rangle_1$ with probability $|1/\sqrt{2}|^2 = 1/2$. In other words, we would know absolutely nothing about the position of the particle!

As a matter of convenience, unless otherwise specified, if basis elements do not have subscripts, we will assume that they refer to basis 1.

2.1 Quantum Measurements

A quantum state sent from Alice to Bob is completely useless to Bob, informationally speaking, unless he can do something which allows him to determine its value. This "something" is a process called quantum measurement, which we briefly describe here. A more complete introduction can be found in Chapter 2 of [8].

Quantum measurement is performed by a collection $\{M_m\}$ of measurement operators that satisfy the *completeness equation*, that is,

$$\sum_m M_m^\dagger M_m = I.$$

The subscript m represents the outcome that may occur. When a measurement operator M_m is applied to a state $|\psi\rangle$, one of two things will happen: either outcome m will occur, or it will not. The probability that outcome m will occur is given by

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle,$$

where we use the notation $\langle\psi|$ to indicate $|\psi\rangle^\dagger$. Satisfaction of the completeness equation assures that all of the probabilities sum to 1. Now when a particular M_m is applied to $|\psi\rangle$,

it maps the state of the system from $|\psi\rangle$ to

$$|\psi'\rangle = \frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}.$$

The square root in the denominator is a normalization factor, as all qubits have norm 1, i.e. $\langle\psi|\psi\rangle = \langle\psi'|\psi'\rangle = 1$. As can be clearly seen, when a state is measured to obtain information, it is rarely left undisturbed. In fact, the only instance in which a measurement operator does not disturb the state of a system is when $|\psi\rangle$ is an eigenstate of the operator M_m. If Bob has no prior information about the state of $|\psi\rangle$, then he could not choose operators in such a way as to guarantee that the state remain undisturbed. On the contrary, in order to determine the maximum amount of information from $|\psi\rangle$, Bob typically chooses operators which leave no other choice but to map $|\psi\rangle$ to some seemingly obscure state.

As an example, consider a state $|\psi\rangle$ prepared in basis 1, i.e. $|\psi\rangle = \alpha_0|0\rangle_1 + \alpha_1|1\rangle_1$. We can use the two measurement operators

$$M_0 = |0\rangle_1\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } M_1 = |1\rangle_1\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Notice that these two operators together satisfy the completeness equation. Moreover, the probability of obtaining a 0 is given by

$$p(0) = \langle\psi|M_0^\dagger M_0|\psi\rangle = |\alpha_0|^2,$$

while the probability of obtaining outcome 1 is given by

$$p(1) = \langle\psi|M_1^\dagger M_1|\psi\rangle = |\alpha_1|^2.$$

If Alice sent many copies of the state $|\psi\rangle$ to Bob and no errors occurred before he measured each state using the above operators, then Bob could eventually estimate the state with high confidance.

Suppose, instead, that Alice encoded the same information into a new state that was encoded with respect to basis 2, that is, $|\phi\rangle = \alpha_0|0\rangle_2 + \alpha_1|1\rangle_2$. Then again assume the state was not changed during transmission, and suppose Bob used the same two operators to measure this state. In this case, the probability of obtaining a 0 is given by

$$p(0) = \langle\phi|M_0^\dagger M_0|\phi\rangle = \frac{1}{2}|\alpha_0 + \alpha_1|^2,$$

while the probability of obtaining a 1 is given by

$$p(1) = \langle\phi|M_1^\dagger M_1|\phi\rangle = \frac{1}{2}|\alpha_0 - \alpha_1|^2.$$

This example shows why it is important for Bob to know in which basis a state was prepared. M_0 and M_1 were chosen to obtain the information encoded in basis 1 of a state. If Bob has no idea in which basis Alice prepared a state, then he has no way of knowing what those probabilities indicate.

2.2 Quantum Noise and the Pauli Operators

Quantum information has many more properties than classical information. These properties open the doorway to new and amazing results (take Shor's algorithm [9], for example); however, these properties also lead to additional difficulties which need not be considered when working with strictly classical information.

In effect, any $2n \times 2n$ complex matrix can describe a noise operation on an n-qubit state. As previously mentioned, measurements generally disturb a state, so any attempt at determining the effects of an error on a corrupted state will not be very useful. Because of

this, we are also unable to clone the corrupted state [10,11] and analyze the cloned state to determine the effects on the original state.

Before discussing methods to correct errors, we will first study some properties of quantum noise. Note that as a matter of convention, we will assume that noise always acts independently on each qubit. There is a special set of operator matrices over $\mathcal{H}_2$ called the *Pauli operators*:

$$\mathsf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \mathsf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \mathsf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \mathsf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Notice that $\mathsf{X}^2 = \mathsf{Y}^2 = \mathsf{Z}^2 = \mathsf{I}$, and that $\mathsf{ZX} = i\mathsf{Y}$. Another important characteristic is that these operators form a basis over $M_2(\mathbb{C})$, the complex vector space of 2×2 matrices.

In particular, notice that $\mathsf{X}|0\rangle_1 = |1\rangle_1$ and $\mathsf{X}|1\rangle_1 = |0\rangle_1$. X is called the "bit-flip" operator on basis 1. Z acts similarly on basis 2. Notice that $\mathsf{Z}|0\rangle_1 = |0\rangle_1$ but $\mathsf{Z}|1\rangle_1 = -|1\rangle_1$. We say that Z is the "phase-flip" operator on basis 1. X acts similarly on basis 2. Given this example, and that $\mathsf{ZX} = i\mathsf{Y}$, it is easy to see how Y acts on a state.

Now suppose we want to protect a single qubit against an error. Recall that a qubit can be written as an element of $\mathcal{H}_2$. An error in this space is simply some 2×2 matrix operating on the qubit. Although there are an infinite number of error operators, we have the convenience of interpreting each one as a linear combination of Pauli matrices. Moreover, since the quantum states are always normalized, and global phases are indistinguishable, we can reduce ourselves to looking at error elements E in the convex closure of the Pauli operators. In other words, if E is some arbitrary error on a single qubit, then we can always rewrite E as $\alpha_0\mathsf{I} + \alpha_1\mathsf{X} + \alpha_2\mathsf{Y} + \alpha_3\mathsf{Z}$, where $|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 = 1$.

Using methods similar to those described in the previous section, measuring the operator E will always collapse it out of a superposition of Pauli operators down to I, X, Y, or Z. In other words, if $E = \alpha_0\mathsf{I} + \alpha_1\mathsf{X} + \alpha_2\mathsf{Y} + \alpha_3\mathsf{Z}$, then measuring the corrupted state $E|\psi\rangle$ with some measurement operator M_m will collapse the state to $\mathsf{I}|\psi\rangle$, $\mathsf{X}|\psi\rangle$, $\mathsf{Y}|\psi\rangle$, or $\mathsf{Z}|\psi\rangle$ with

probabilities $|\alpha_0|^2$, $|\alpha_1|^2$, $|\alpha_2|^2$, and $|\alpha_3|^2$, respectively. Because of this, when considering errors on a single qubit, it suffices to consider only those errors $E \in \{\mathsf{I}, \mathsf{X}, \mathsf{Y}, \mathsf{Z}\}$. Thus, in order to correct a completely arbitrary error on a single qubit, we need only to be able to correct this finite set of errors.

2.3 Pauli Operators and the Pauli Group

If we include the scalars ± 1 and $\pm i$ with the set of Pauli operators, then we get a multiplicative group, called the *Pauli group* on a single qubit, denoted G_1; that is, $G_1 = \{\pm\mathsf{I},\ \pm i\mathsf{I},\ \pm\mathsf{X},\ \pm i\mathsf{X},\ \pm\mathsf{Y},\ \pm i\mathsf{Y},\ \pm\mathsf{Z},\ \pm i\mathsf{Z}\}$. As just explained, any arbitrary error on a single qubit state can be viewed as an element of G_1. Note that ± 1 and $\pm i$ act as global phases, and so are physically indistinguishable; however, we include them here so that we can use the group-theoretic properties to assist us. Note that every element of this group is unitary (i.e. $E^\dagger = E^{-1}$), and either Hermitian or skew-Hermitian (i.e. $E^\dagger = \pm E$).

This group can be generalized naturally to act on n-qubit states by considering n-fold tensor products of the elements of G_1. For example, the Pauli group on two qubits is given by

$$\begin{aligned} G_2 = \{&\pm\mathsf{II}, \pm i\mathsf{II}, \pm\mathsf{IX}, \pm i\mathsf{IX}, \pm\mathsf{IY}, \pm i\mathsf{IY}, \pm\mathsf{IZ}, \pm i\mathsf{IZ}, \\ &\pm\mathsf{XI}, \pm i\mathsf{XI}, \pm\mathsf{XX}, \pm i\mathsf{XX}, \pm\mathsf{XY}, \pm i\mathsf{XY}, \pm\mathsf{XZ}, \pm i\mathsf{XZ}, \\ &\pm\mathsf{YI}, \pm i\mathsf{YI}, \pm\mathsf{YX}, \pm i\mathsf{YX}, \pm\mathsf{YY}, \pm i\mathsf{YY}, \pm\mathsf{YZ}, \pm i\mathsf{YZ}, \\ &\pm\mathsf{ZI}, \pm i\mathsf{ZI}, \pm\mathsf{ZX}, \pm i\mathsf{ZX}, \pm\mathsf{ZY}, \pm i\mathsf{ZY}, \pm\mathsf{ZZ}, \pm i\mathsf{ZZ}\}, \end{aligned}$$

where we use the notation AB in the sanserif typface to indicate $A \otimes B$.

Note that if $\mathsf{A}_1\mathsf{A}_2 \ldots \mathsf{A}_n \in G_n$, then $\mathsf{A}_1\mathsf{A}_2 \ldots \mathsf{A}_n$ acting on an n-qubit state $|a_1 a_2 \ldots a_n\rangle =$

$|a_1\rangle \otimes |a_2\rangle \otimes \cdots \otimes |a_n\rangle$ gives us

$$\mathsf{A}_1\mathsf{A}_2 \ldots \mathsf{A}_n|a_1 a_2 \ldots a_n\rangle = \mathsf{A}_1|a_1\rangle \otimes \mathsf{A}_2|a_2\rangle \otimes \cdots \otimes \mathsf{A}_n|a_n\rangle.$$

In other words, using this notation allows us to easily identify which operator is acting on which qubit.

There are a few observations and key properties about this group that will be of great importance later. The first is that any two elements of G_n will always either commute or anticommute; that is, given $E_i, E_j \in G_n$, then $E_i E_j = \pm E_j E_i$ (this is easily seen to be true by inspection in the case of G_1, and is proved by induction for arbitrary n). By induction, we can also show that all elements in G_n are unitary, and either Hermitian or skew-Hermitian.

Chapter 3: Quantum Stabilizer Codes

The stabilizer codes are a class of quantum error correcting codes that have beautiful algebraic properties. They were first discovered by Gottesman [12], and grew in popularity very rapidly once it was shown that they could be described classically [13].

3.1 The Stabilizer Group

As we previously explained, if the action of elements of G_n on n-qubit states can be corrected, then arbitrary errors on n-qubit states can be corrected. Let $\mathcal{S} \leq G_n$ be a non-trivial abelian subgroup of G_n such that $-I_n \notin \mathcal{S}$, and suppose $\mathcal{S}$ has generating set $\mathcal{G} = \{g_1, g_2, \ldots, g_{n-k}\}$. We call $\mathcal{S}$ the *stabilizer group* for a stabilizer code $C(\mathcal{S})$, where the stabilizer code $C(\mathcal{S})$ is the $+1$ eigenspace of $\mathcal{S}$, i.e. collection of states $|\psi\rangle$ in $\mathcal{H}_2^n$ satisfying

$$S|\psi\rangle = |\psi\rangle, \forall\ S \in \mathcal{S}.$$

We say these states are *stabilized* by every element of $\mathcal{S}$. For example, suppose $\mathcal{S} = \langle \mathsf{ZZI}, \mathsf{IZZ} \rangle \leq G_3$,the subgroup generated by the states ZZI, and IZZ. Then the $+1$ eigenspace $C(\mathcal{S})$ is spanned by the states $|000\rangle$ and $|111\rangle$.

Note that we could just as easily start with a subspace in $\mathcal{H}_2^n$, calling this the stabilizer code C. The stabilizer group $\mathcal{S}(C)$ would then be defined as the unique subgroup $\mathcal{S}$ of G_n not containing $-I_n$ for which C is the $+1$ eigenspace. Note that it is straightforward to show that this is in fact an abelian subgroup of G_n. However, starting with a stabilizer group $\mathcal{S}$, we find the code $C(\mathcal{S})$ as the unique $+1$ eigenspace of $\mathcal{S}$. In other words, $\mathcal{S}(C(\mathcal{S})) = \mathcal{S}$. As it turns out, constructing a code from its stabilizer group is generally more preferable, as many of the properties of the code are directly determined by the stabilizer group.

3.2 Errors and Stabilizers

Suppose $\mathcal{S} \leq G_n$ is a stabilizer group with generating set $\mathcal{G} = \{g_1, g_2, \ldots, g_{n-k}\}$ and corresponding stabilizer code $C(\mathcal{S})$. Now suppose $|\psi\rangle \in C(\mathcal{S})$ is transmitted, and gets acted on by some error $E \in G_n$.

What happens when $E \in \mathcal{S}$? In this case, E does nothing to $|\psi\rangle$. What about when $E \in G_n - Z(\mathcal{S})$ (where $Z(\mathcal{S})$ is the *centralizer* of $\mathcal{S}$, the set of all elements in G_n that commute with every element in $\mathcal{S}$)? In this case, there exists at least one element of $\mathcal{S}$ (and hence some element of the generating set) that does not commute with E, say $g_i E|\psi\rangle = -Eg_i|\psi\rangle = -E|\psi\rangle$. The ± 1 value that we obtain from this operation is in fact a measurable value (the details of why we can obtain this lie outside the scope of this paper, and we refer the reader to [8] for more details). When we get a -1, we detect that an error did, in fact, occur. Operating on $E|\psi\rangle$ sequentially by each of the elements of $\mathcal{G}$ gives us a sequence of ± 1's that in effect act as a syndrome to allow us to determine the exact error that occurred.

The real problem occurs when $E \in Z(\mathcal{S}) - \mathcal{S}$. In this case, E will commute with each element of $\mathcal{S}$, and will thus give the same syndrome as would an element of $\mathcal{S}$ operating on the state. Hence, the elements in $Z(\mathcal{S}) - \mathcal{S}$ are all undetectable errors.

3.3 Error Correction Conditions for Stabilizer Codes

In this section, we give the formal theorem stating the types of errors that can be corrected by a stabilizer code. Rather than proving the theorem here (this is from Theorem 10.8 in [8], which contains a proof), we motivate it with an example.

Suppose E_j is some error acting on the code space of a stabilizer code. Then the error syndrome is given by β_l such that $E_j g_l E_j^\dagger = \beta_l g_l$, where $g_l \in \mathcal{G}$ is a generating element of $\mathcal{S}$. If E_j is the unique operator with this syndrome, then we can correct for it simply by applying $E_j^\dagger$.

If there exists an $E_k \neq E_j$ having the same syndrome, it follows that $E_j g_l E_j^\dagger = E_k g_l E_k^\dagger$, and hence $E_j^\dagger E_k g_l E_k^\dagger E_j = g_l$. Since $g_l \in \mathcal{S}$, then $E_j^\dagger E_k$ is in $Z(\mathcal{S})$. If, in fact, $E_j^\dagger E_k$ is in $\mathcal{S}$, then applying $E_j^\dagger$ after the error E_k occurs will still result in a successful recovery. This brings us to the theorem, from [8].

Theorem 3.1 (Error correcting conditions for stabilizer codes)**.** Let $\mathcal{S}$ be a stabilizer for a stabilizer code $C(\mathcal{S})$. Suppose $\{E_j\}$ is a set of operators in G_n such that $E_j^\dagger E_k \notin N(\mathcal{S}) - \mathcal{S}$ for all j, k. Then $\{E_j\}$ is a correctable set of errors for the code $C(\mathcal{S})$.

Here, $N(\mathcal{S}) := \{E \in G_n : ES_iE^\dagger \in \mathcal{S} \; \forall \; S_i \in \mathcal{S}\}$ is called the *normalizer* of $\mathcal{S}$. Under the assumption that $-I_n \notin \mathcal{S}$, it can be shown that $N(S) = Z(S)$, and so the theorem makes sense with our example.

3.4 Minimum Weight of a Stabilizer Code

We now know which types of errors go undetected when acting on a stabilizer code. In order to obtain some quantifiable notion of "how well" a given stabilizer code works, it would be preferable to know how many independent, arbitrary qubit errors on a single state can be corrected. This motivates our definition of the *minimum weight* of a quantum stabilizer code.

Define the *weight* of an element of G_n to be the number of non-identity terms in the tensor product. E.g., if $E = \mathsf{IIXYZZI} \in G_7$, then the weight of E, denoted $wt(E)$, is 4. Define the *minimum weight* of a stabilizer code $C(\mathcal{S})$ to be $\min\{wt(E) \mid E \in Z(\mathcal{S}) - \mathcal{S}\}$, the smallest weight of any element of $Z(\mathcal{S}) - \mathcal{S}$.

This notion of minimum weight is, by construction, very similar to the notion of minimum distance of a classical code. If a given stabilizer code $C(\mathcal{S})$ has minimum weight d, then all elements of G_n having weight less than d are either in $\mathcal{S}$ or in $G_n - Z(\mathcal{S})$. In other words, all operators in G_n of weight less than d are identifiable errors. Similarly to classical codes, a stabilizer code with minimum weight d can detect and correct up to $t = \lfloor \frac{d-1}{2} \rfloor$ and

fewer arbitrary errors.

Chapter 4: Classical Representation of Stabilizers

We will now discuss a method to classically represent a quantum stabilizer code in a manner similar to classical codes. In fact, as it turns out, there is a large class of stabilizer codes, called CSS codes (named after their discoverers, Calderbank, Shor [14], and Steane [15]), which can be constructed from classical error correcting codes. This class of codes is well studied, and a nice introduction can be found in Section 10.5 of [8]. We will discuss the general classical stabilizer constructions, and then end this section with some observations about CSS constructions.

4.1 The Pauli Vector

In order to understand the classical representation, which was discovered by Calderbank, Rains, Shor, and Sloane [13], we first make some observations about the group G_n, and their implications. Note that the center of G_n is given by $C_{G_n} = \{i^k I : k \in \mathbb{Z}_4\}$, where I is the identity matrix in G_n. We want to consider the group $\overline{G_n} = G_n/C_{G_n}$. Note that its elements are equivalence classes of the form $\overline{E} = \{i^k E : k \in \mathbb{Z}_4\}$, where $E \in G_n$. Now every element in $\overline{G_n}$ has order 2, making this an *elementary abelian group*, and hence a vector space over $\mathbb{Z}_2$. In fact, we can describe any $\overline{E} \in \overline{G_n}$ as a concatenation of two length n binary vectors called the X and Z vectors of $\overline{E}$ in the following way.

Let $\overline{E} \in \overline{G_n}$ be represented by the element in its equivalence class having a $+1$ coefficient, which we will call the *scalar free element*. We place a 1 in position i of the X vector whenever there is an X or Y in position i of $\overline{E}$. Similarly, we place a 1 in position j of the Z vector whenever there is a Z or Y in position j of $\overline{E}$. The Y values are listed in both vectors since Y is a product of X and Z.

If $\overline{E} \in \overline{G_n}$ has X vector E_X and Z vector E_Z, then the concatenated vector $(E_X|E_Z)$ is called the *Pauli vector* for $\overline{E}$, or simply the Pauli $\overline{E}$-vector. For example, IXXYZIZ has the corresponding Pauli vector (0111000|0001101). If we instead concatenate as $(E_Z|E_X)$, we call this the *reversed Pauli vector* for $\overline{E}$, or simply the reversed $\overline{E}$-vector.

It can be shown [13] that two elements $S, T \in G_n$ commute if and only if the corresponding Pauli vectors $(S_X|S_Z)$ and $(T_X|T_Z)$ of their images $\overline{S}$ and $\overline{T}$ in $\overline{G_n}$ (respectively), satisfy the *twisted inner product* requirement, namely

$$ST = TS \iff \langle S_X, T_Z \rangle + \langle S_Z, T_X \rangle = 0, \tag{4.1}$$

where $\langle \cdot, \cdot \rangle$ indicates the standard inner (dot) product over the n-dimensional binary vector space V_n, and addition is performed modulo 2. Notice that this is equivalent to saying that S and T commute if and only if the reversed $\overline{S}$-vector is orthogonal to the $\overline{T}$-vector under the standard inner product over the $2n$-dimensional binary vector space V_{2n}, i.e.

$$ST = TS \iff \langle (S_Z|S_X), (T_X|T_Z) \rangle = 0. \tag{4.2}$$

Note that previously, we had defined the stabilizer group $\mathcal{S}$ as an abelian subgroup of G_n that did not contain the element $-I_n$. It can further be shown that if $-I_n \notin \mathcal{S}$, then neither is $\pm iI_n$. But then each element of $\mathcal{S}$ will be in a different equivalence class in $\overline{G_n}$. It follows that we can view each element of $\mathcal{S}$ as a label for its corresponding equivalence class in $\overline{G_n}$. Recalling from earlier that the scalars ± 1 and $\pm i$ act as global phases, and are hence undetectable, it actually makes sense to view errors as elements of $\overline{G_n}$ rather than the larger group G_n. In other words, there is no loss of generality by viewing all errors as elements of $\overline{G_n}$ rather than G_n as we had done previously. Note that C_{G_n} is not a subgroup of $\mathcal{S}$, so $\overline{\mathcal{S}} \neq \mathcal{S}/C_{G_n}$ (in fact, $\overline{\mathcal{S}} = \mathcal{S}C_{G_n}/C_{G_n}$). However, while the map $G_n \to G_n/C_{G_n}$ is a surjective homomorphism, the restriction of this map to $\mathcal{S}$ is in fact injective. Because of this we will use both $\mathcal{S}$ and $\overline{\mathcal{S}}$ interchangeably, calling both the stabilizer group. By

convention, the elements of $\overline{\mathcal{S}}$ are represented by the scalar free elements of each of the equivalence classes in $\overline{\mathcal{S}}$.

4.2 Classical Representation of Error Correction

Let $\mathcal{S} \leq G_n$ be a stabilizer group, with generating set $\mathcal{G} = \{g_1, g_2, \ldots, g_{n-k}\}$. We construct the *stabilizer parity check* matrix $H = [H_X|H_Z]$ by letting the rows of H correspond to the Pauli vectors of the elements of $\overline{\mathcal{S}}$. Because the elements of $\mathcal{S}$ commute by definition, it follows that the rows of H will satisfy the twisted inner product requirement (4.1). Conversely, if A and B are binary $n \times (n-k)$ matrices such that any two rows in the concatenated matrix $[A|B]$ satisfy the twisted inner product, then $[A|B]$ corresponds to a quantum stabilizer code; the rows of the concatenated matrix correspond to elements of $\overline{G_n}$, and their span corresponds to a subspace $\overline{\mathcal{S}}$ of $\overline{G_n}$, as an elementary abelian group, in which any two elements are orthogonal with respect to the twisted inner product; that is, $\overline{\mathcal{S}}$ is a stabilizer group.

The requirement that any two rows of a concatenated binary matrix $[A|B]$ satisfy the twisted inner product is equivalent to saying that $AB^T + BA^T = 0$, the all zero matrix. In other words, if two binary $n \times (n-k)$ matrices A and B satisfy $AB^T + BA^T = 0$, then $[A|B]$ is a parity check matrix for a quantum stabilizer code. Moreover, an element E with image $\overline{E}$ in $\overline{G_n}$ commutes with each stabilizer for this stabilizer code precisely when the reversed $\overline{E}$-vector is orthogonal to each row of the parity check. In other words, E commutes precisely when $[A|B](E_Z|E_X)^T = 0$, the zero vector.

Suppose E is a correctable error acting on the state $|\psi\rangle \in C(\mathcal{S})$. To determine the error, we operate on it with each element of the generating set $\mathcal{G}$ of $\mathcal{S}$ to obtain a syndrome. We simulate this classically by letting H be the parity check matrix for the stabilizer code, and representing E with its error vector $(E_Z|E_X)$. We then perform the operation $H(E_Z|E_X)^T$ to obtain the classical syndrome. We obtain the all zero syndrome whenever E commutes with each element of $\mathcal{G}$, and so we assume in this case that no error occurs.

In other words, if $\{(S_Z|S_X)_j\}$ is a collection of error vectors corresponding to the stabilizers for a stabilizer code, then $\{(S_Z|S_X)_j\} \subseteq \ker(H)$; that is, they are a subset of the kernel of H as an action on the binary vector space V_{2n}. In fact, given a quantum stabilizer parity check matrix H, its kernel precisely corresponds to the centralizer $Z(\mathcal{S})$ of the corresponding stabilizer code.

Suppose $H = [A|B]$ is a length $2n$ (i.e. A and B each have n columns) stabilizer parity check matrix (i.e. $AB^T + BA^T = 0$) having dimension $n - k$. Then $\ker(H)$ will have dimension $n + k$. Moreover, H will correspond to a stabilizer code that encodes k qubits into n qubits. If this code has minimum weight d, we call such a code a quantum $[[n, k, d]]$ code.[1]

While this classical representation greatly aids our ability to understand and analyze stabilizer codes, finding parity check matrices that correspond to quantum stabilizer codes having good performance parameters (i.e. $n-k$ is small and d is large) is no trivial matter. The CSS construction of quantum stabilizer codes has offered great promise, however, by allowing us to use classical codes with known parameters to quickly determine the performances of their quantum counterparts.

4.3 CSS Constructions

CSS codes are a class of quantum stabilizer codes constructed from classical error correcting codes. These codes are particularly nice in that many of the parameters of a classical error correcting code carries over to the corresponding CSS code. More formally, suppose $\mathcal{C}_1$ is a classical binary linear $[n, k_1, d_1]$ code, and that $\mathcal{C}_2 \subset \mathcal{C}_1$ such that $\mathcal{C}_2^\perp$ is a classical binary linear $[n, k_2, d_2]$ code. Let $H(\mathcal{C}_1)$ and $H(\mathcal{C}_2^\perp)$ be parity check matrices for $\mathcal{C}_1$ and $\mathcal{C}_2^\perp$

[1] It is a relatively standard practice to place the parameters of a quantum code in double brackets in order to distinguish them from the classical parameters.

respectively. Then the matrix of the form

$$H(C) = \begin{bmatrix} H(\mathcal{C}_2^{\perp}) & 0 \\ 0 & H(\mathcal{C}_1) \end{bmatrix}$$

is a parity check matrix for a CSS code. Note that, since $\mathcal{C}_2 \subset \mathcal{C}_1$, it follows that $H(\mathcal{C}_2^{\perp})H(\mathcal{C}_1)^T = 0$, and hence, the twisted inner product requirement will be satisfied. Moreover, the corresponding CSS code C with parity check matrix $H(C)$ above will be an $[[n, K, D]]$ quantum stabilizer code, where $K = k_1 + k_2 - n$ and $D \geq \min(d_1, d_2)$, i.e., C will encode K qubits into n qubits, and will correct arbitrary errors on up to $t = \lfloor \frac{D-1}{2} \rfloor$ and fewer qubits. In fact, if A and B are two $n \times (n-k)$ binary matrices, then

$$\begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}$$

is a CSS code if and only if $AB^T = 0$.

Suppose $\overline{\mathcal{S}}$ is the stabilizer group for the code with parity check matrix given by $H(C)$, and with corresponding centralizer $Z(\overline{\mathcal{S}})$. Note that the space orthogonal to $H(C)$ (with respect to the twisted inner product) corresponds to $Z(\overline{\mathcal{S}})$. But then the distance $D = \min(d_1, d_2)$ actually corresponds to the minimum weight of $Z(\overline{\mathcal{S}})$. Recall that the minimum distance of a stabilizer code is given by the minimum weight of $Z(\overline{\mathcal{S}}) - \overline{\mathcal{S}}$, and so D is actually a lower bound on the minimum distance of the code.[2]

A particularly appealing CSS construction is one in which the dual $\mathcal{C}^{\perp}$ of a classical code $\mathcal{C}$ is a subset of the code, i.e., $\mathcal{C}^{\perp} \subset \mathcal{C}$. In this case, if $\mathcal{C}$ is an $[n, k, d]$ code with parity

[2]Only knowing the lower bound is generally not a problem, as this is typically the case for many classical error correcting codes. In particular, BCH codes are an example of a very large class of classical codes constructed to have a fixed lower bound on their minimum distance.

check matrix $H(\mathcal{C})$, then

$$H(C) = \begin{bmatrix} H(\mathcal{C}) & 0 \\ 0 & H(\mathcal{C}) \end{bmatrix}$$

is a parity check for a quantum $[[n, 2k-n, D]]$ stabilizer code C, where $D \geq d$. The twisted inner product requirement is again satisfied by the fact that $\mathcal{C}^{\perp} \subset \mathcal{C}$, and hence $H(\mathcal{C})H(\mathcal{C})^T = 0$. CSS codes having this particular construction are called *symmetric* codes; otherwise they are called *asymmetric* CSS codes.

4.4 Example: Seven Qubit Steane Code

In order to better understand the CSS construction, we will use the example of a well-known CSS code discovered by Steane [16], one of the members after whom CSS codes are named. Generators for the stabilizer group for the seven qubit Steane code S_7 are given by

IXXXXII

XIXXIXI

XXIXIIX

IZZZZII

ZIZZIZI

ZZIZIIZ.

One can quickly observe that all of these generators commute with one another. Now, the classical representation gives the following parity check matrix:

$$\left[\begin{array}{ccccccc|ccccccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right].$$

This parity check matrix takes the form

$$\begin{bmatrix} H(\mathcal{C}) & 0 \\ 0 & H(\mathcal{C}) \end{bmatrix},$$

where $H(\mathcal{C})$ is self-orthogonal, showing that this is, in fact, a symmetric CSS code. Moreover, observing that $H(\mathcal{C})$ is the parity check matrix from (1.1) for the classical Hamming $[7, 4, 3]$ code, we can conclude that S_7 is a $[[7, 1, 3]]$ quantum stabilizer code, encoding 1 qubit into 7 qubits, and protecting against arbitrary single-qubit errors.

Using the CSS constructions, we can find quantum stabilizer codes with relatively high performance. In particular, low-density parity check (LDPC) codes are some of the best known classical codes. They, like quantum stabilizer codes, are constructed from their parity check matrices. However, as we will show later, constructing LDPC codes that satisfy the quantum stabilizer conditions is nontrivial. We will take a slight detour here to discuss finite projective planes, and how they are used to construct exceptionally high performing classical LDPC codes. We then show how they can be modified to construct some of the best-known quantum LDPC codes in the current literature.

Chapter 5: Finite Projective Planes

A finite projective plane π is a finite collection of points, along with subsets of points (lines), which satisfy the following three axioms:

- Any two distinct points determine a unique line.
- Any two distinct lines determine a unique point.
- There exist four points, no three of which are colinear.

The third axiom exists simply to eliminate trivial examples. Notice that the second axiom implies that there are no parallel lines. The simplest example of a finite projective plane is called the *Fano plane*, a construction of which is depicted in Figure 5.1.

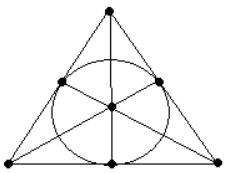

Figure 5.1: The Fano Plane, $PG(2,2)$

Notice that the Fano plane has 7 points and 7 lines. Each line is made up of 3 points, and each point has 3 incident lines. This symmetry is no coincidence. Suppose there is a line in a given projective plane that has exactly $q+1$ distinct points. It can be shown that every line will then contain exactly $q+1$ points, and we call q the *order* of the given

projective plane π. The following useful properties of a projective plane of order q can be determined using basic counting techniques:

- Every line contains $q+1$ distinct points.
- Every point is incident with $q+1$ distinct lines.
- π has q^2+q+1 distinct points and q^2+q+1 distinct lines.

The Fano plane from Figure 5.1 is a projective plane of order $q=2$. In this paper, we will be focusing on projective planes that are *2-dimensional projective geometries* of order $q=p^s$ for some prime p, often using the notation $PG(2,q)$ to refer to such geometries.[1] The projective plane $\pi = PG(2,q)$ has nice algebraic properties and coordinate constructions that we will rely on later.

We can construct an incidence matrix M_π for π by letting the lines in π correspond to the rows in M_π, and the points correspond to columns. There is a 1 in position i,j of M_π precisely when line i contains point j, and a 0 otherwise. Notice that, because of the first two properties given above, the incidence matrix for π is sparse by construction. Additionally, we can let the rows denote bit nodes and columns denote check nodes in a Tanner graph. It follows from the first two requirements of a finite projective plane that there will be no four-cycles in the graph. Because of this, projective planes make great candidates for constructions of LDPC codes. Moreover, it was shown by Singer [17] that when $\pi = PG(2,q)$ where q is a power of a prime, then M_π can be cyclically generated (i.e., each of its rows are cyclic shifts of the same vector), and hence the corresponding code is cyclic, allowing it to be easily encoded from its generator polynomial in a linear feedback shift register.[2] This method of constructing classical LDPC codes from the incidence structures of finite geometries was first shown in 2001 [18]. We will now show how M_π can be adapted to construct our first example of a parity check for a quantum LDPC code.

[1] In fact, the existance of projective planes of non-prime power order is still an open question.

[2] Explaining what all of this means lies outside the scope of this paper. The point to take from this is that codes constructed in this manner can be easily generated and require virtually no memory on a computer.

5.1 Projective Planes and Quantum LDPC Codes

The very aspect of projective planes that makes them so nice in classical codes now poses a problem for quantum codes. Notice that any two rows of the incidence matrix M_π of a projective plane $\pi = PG(2, q)$ have exactly one 1 in common (or, equivalently, we say that any two rows *overlap* exactly once). This result eliminates any chance of four cycles in the corresponding Tanner graph. Suppose we want to use M_π to construct a CSS code. We cannot do the simple method of constructing the CSS code having the form

$$\begin{bmatrix} M_\pi & 0 \\ 0 & M_\pi \end{bmatrix}$$

since M_π is never self-orthogonal. As can be seen, the twisted inner product requirement of quantum stabilizer codes forces four-cycles on the Tanner graph. We can adapt M_π, however, to satisfy this requirement by making it self-orthogonal. This is done by adding a column of all ones at the end of the matrix, calling the new matrix M'_π. Since any two rows of M'_π overlap exactly twice, any two rows are orthogonal, and hence the twisted inner product requirement is satsified.

Now consider $\pi = PG(2, 2^s)$ for some positive integer s. In this case, every row of M_π has an odd number of "ones." But then adding (modulo 2) all of the columns in M_π gives the all ones vector. Thus, the column of all ones is linearly dependent on the other columns, and hence adding it to M_π will not affect the rank of the matrix, i.e. $rank(M_\pi) = rank(M'_\pi)$. This result becomes rather important to us, so we will state it here, in the form of a lemma, for ease of reference.

Lemma 5.1. *Suppose M is a matrix generating a subspace of a binary vector space, and suppose further that each row of M has an odd number of "ones." Let M' be the matrix formed by adding a column of all ones to M. Then $rank(M) = rank(M')$.*

Now, in the case of $\pi = PG(2, 2^s)$, we have the following result.

Theorem 5.2. Let $\pi = PG(2, 2^s)$, and let M'_π be the incidence matrix of π with a column of all ones attached to the end. Then the matrix

$$H(C_\pi) = \begin{bmatrix} M'_\pi & 0 \\ 0 & M'_\pi \end{bmatrix}$$

is a parity check matrix for a quantum $[[n, 2k-n, D]]$ stabilizer code C_π, where $n = 4^s+2^s+2$, $k = 4^s - 3^s + 2^s + 1$, and $D = 2^s + 2$.

Proof. Clearly, $n = 4^s + 2^s + 2$. It was shown [19] that when $q = 2^s$, we have that $rank(M_\pi) = 3^s+1$, and thus by Lemma 5.1, $rank(M'_\pi) = 3^s+1$. It follows that the classical code with parity check matrix M'_π has dimension $k = n - rank(M'_\pi) = 4^s - 3^s + 2^s + 1$, and as discussed in Section 4.3, the CSS code will have dimension $2k - n$. In order to determine the minimum weight, first observe that the minimum weight of C_π is the same as the minimum weight of the classical code having parity check matrix given by M'_π. We now separate the problem into two cases, namely, the case in which a minimum weight vector v has a 0 at the end and the case when it has a 1. If v has a 0 at the end, then it will be a copy of a codeword of the classical code with parity check matrix M_π, with an additional zero at the end. The minimum weight of this code is known to be $2^s + 2$ [18].

Now suppose $v_n = 1$, that is, the last bit of v is a 1. If v_n were the only 1 bit in v, then v would not be orthogonal to any row in M'_π. Suppose $v_i = 1$ where $i < n$. Then v is orthogonal to exactly $2^s + 1$ of the $4^s + 2^s + 1$ rows. Adding another 1 to v will cause v to be orthogonal to another $2^s + 1$ rows of M'_π. Now some of the rows that were previously orthogonal may no longer be orthogonal to v, but let us assume the best case that each time we add a 1 to v we do not destroy the orthogonality of the previously orthogonal rows. This will provide us with a lower bound. Then if $v_n = 1$, we must have at least m additional 1's, where m is the smallest integer such that $m(2^s + 1) \geq 4^s + 2^s + 1$. Thus $m = 2^s + 1$, giving us the lower bound of $2^s + 2$ on the weight of v if $v_n = 1$. □

Note that $H(C_\pi)$ is sparse by construction, giving us our first example of a type of quantum low-density parity check (QLDPC) code constructed from the projective plane $\pi = PG(2, q)$. This code has a rate[3] that increases rapidly with n, and a minimum weight greater than $\sqrt{n}$. Adding the column of all ones does, however, have the negative effect of creating four-cycles in the corresponding Tanner graph, since any two distinct rows of M'_π will overlap in exactly two points (i.e. have two "ones" in common position). Unfortunately, in the construction of symmetric CSS codes, any two distinct rows of the parity check must overlap in an even number of places in order to satisfy the twisted inner product requirement. Thus, each pair of rows overlapping exactly twice is actually a best-case scenario in the case of symmetric CSS codes. Another potentially negative fact is the rather large number of check nodes in this Tanner graph. While extra check nodes may benefit the iterative decoding algorithm, it does force more operations on the quantum state than absolutely necessary.

In an effort to reduce the number of check nodes in the Tanner graph of a QLDPC code, while maintaining no more than a single four-cycle between any two rows of the parity check, we can instead turn to asymmetric CSS code constructions based on the projective plane. We will show that this can be done by first separating the rows of M'_π into two distinct sets, say M'_{π_1} and M'_{π_2}. The matrix

$$\begin{bmatrix} M'_{\pi_1} & 0 \\ 0 & M'_{\pi_2} \end{bmatrix}$$

is constructed, forming the parity check for an asymmetric QLDPC code. This method reduces the total number of check nodes in the Tanner graph in half, while maintaining no more than a single four-cycle between any two check nodes. Moreover, every four-cycle will contain the bit node corresponding to the column of all ones.

[3]The *rate* of a code is a notion of how much information is contained in the codewords. If the code has length n and dimension k, then the rate is given by $\frac{k}{n}$.

The column of all ones, or *unit column*, that was added to the original incidence matrix M_π will be brought up quite frequently in this paper, so for simplicity, we will call it the u-column. When discussing the bit node corresponding to the u-column in the Tanner graph, we call it the u-bit node. The projective plane used to construct M_π will also be discussed. We can interpret the u-column graphically as being a point, which we will call the u-point, added to the projective plane having the property that every line intersects it. Note that this graphical interpretation is technically no longer a projective plane. Since we are adding a point, we will call it the *extended projective plane*, denoted $PG(2,q)'$ or π'.

Choosing how to separate M'_π into M'_{π_1} and M'_{π_2} can be done randomly, but it would then be difficult to determine how well the QLDPC code will perform. The next few sections will describe an alternative method of describing the projective plane, and how this can be used to characterize certain subsets of the plane.

5.2 Coordinate Construction of $PG(2,q)$

While the graphical representation of $\pi = PG(2,q)$ is useful, and will be heavily relied upon to determine later results, it is occassionally necessary to rely on the underlying algebraic structure. In order to do this, we must continue the assumption that $q = p^s$ for some prime p, and then more formally describe points and lines in the general projective geometry $PG(m,q)$ as 1-dimensional and 2-dimensional subspaces (respectively) of $V(m+1,q)$, the $(m+1)$-dimensional vector space over the unique finite field $\mathbb{F}_q$ containing q elements. Since we are focusing specifically on the projective *plane*, points and lines can be described over $V(3,q)$, the 3-dimensional affine vector space over $\mathbb{F}_q$. An effort is made to be complete here, but further details can be found in Chapter 9 of [20].

Since the points in $PG(2,q)$ are 1-dimensional subspaces of $V(3,q)$, each point is given by the equivalence class $\{(cx,cy,cz) \mid c \in \mathbb{F}_q\}$ for fixed x, y, and z, not all zero. We label this point (equivalence class) by $[x,y,z]$, and x, y, and z are called *homogeneous coordinates* for the point. Because of this, we can always uniquely describe each point in $PG(2,q)$ as

an element in $V(3,q)$ of the form $[0,0,1]$, $[0,1,m]$, or $[1,m,n]$, for $m,n \in \mathbb{F}_q$.

Any line in $PG(2,q)$ can be represented by a linear equation $ax+by+cz=0$, where a, b, and c are not all zero. Because constant multiples of this equation will not change the set of points on the line, we can also represent lines as equivalence classes $\{(ma,mb,mc) \mid m \in \mathbb{F}_q\}$ for fixed a, b, and c. We will label the equivalence classes corresponding to lines in parenthesis instead of brackets to distinguish lines from points (i.e. represent $\{(ma,mb,mc) \mid m \in \mathbb{F}_q\}$ by (a,b,c)). Like points, each line can be described uniquely as an element of the form $(0,0,1)$, $(0,1,m)$, or $(1,m,n)$, for $m,n \in \mathbb{F}_q$.

Using this coordinate construction of $PG(2,q)$ provides a clearer method of constructing these planes. It also allows us to more clearly characterize subsets of the planes.

Chapter 6: Subsets of Projective Planes and LDPC Codes

As mentioned previously, the incidence matrix M_π for a projective plane $\pi = PG(2, q)$ can act as a parity check matrix for a classical LDPC code. The parity check need not be square, nor do each of its rows need to be completely linearly independent. Thus we can just as easily look at LDPC parity checks constructed from subsets of points and lines in a finite projective plane. Such work has recently been done, with particular focus on classical LDPC codes constructed from subsets of points and lines in a projective plane relative to nondegenerate conics and regular hyperovals [21, 22]. The results, specifically those from Castleberry et al [22], will be heavily relied upon when constructing QLDPC codes. For the sake of completeness, we give an overview of the methods implemented in the above sources. In particular, we maintain the assumption that $q = 2^s$.

An *arc* in $PG(2, q)$ is a collection of points, no three of which are collinear. It can be shown that there always exists an arc of size $q + 1$ in $PG(2, q)$, and we call this arc an *oval*. There is a particular subset of ovals in $PG(2, q)$ called conics. A *conic* $\mathcal{C}$ is formally defined as a set of points whose coordinates satisfy a non-degenerate quadratic equation, that is,

$$\mathcal{C} := \{[x, y, z] : ax^2 + by^2 + cz^2 + fyz + gzx + hxy = 0\}$$

for some $a, b, c, f, g, h \in \mathbb{F}_q$, where the equation cannot be reduced to an equation of less than 3 variables via linear substitution. Given a conic, we can classify all of the lines in the plane relative to the conic by defining lines to be either *skew* (those that intersect the conic in no points), *secant* (those that intersect the conic in exactly two points), and *tangent* (those that intersect the conic at exactly one point). Every line in the plane falls under one of these classes.

A well-known result in finite projective geometry is that when $q = 2^s$, all of the tangent lines are concurrent at a point outside of the conic, called the *nucleus*. More precisely, if $\mathcal{C}$ is a nondegenerate conic satsifying the quadratic form $ax^2 + by^2 + cz^2 + fyz + gzx + hxy = 0$, then the nucleus will be the point with homogeneous coordinates $[f, g, h]$. Adding the nucleus to the conic gives an arc of size $q + 2$, called a *regular hyperoval* (which will simply be referred to as a *hyperoval* throughout the rest of this report). Since the lines tangent to the conic intersect the hyperoval in exactly two points, it follows that any line in the plane can be distinctly labeled as either skew to the hyperoval or secant to it. It can be shown that, given a hyperoval H, there will always be $(q^2+3q+2)/2$ secant lines to H, and $(q^2-q)/2$ skew lines to H. It can also be shown that any non-hyperoval point is intersected by $(q + 2)/2$ secant lines and $q/2$ skew lines [22]. Unless otherwise specified, we will let $\mathcal{C}$ refer to the conic defined by the quadratic form $y^2 = xz$, and $H_{\mathcal{C}}$ refer to the extension of this conic to a regular hyperoval.

Because the properties of quantum CSS codes can be obtained from the classical codes from which they are constructed, we will spend the next few sections describing classical codes whose parity check matrices are constructed from subsets of points and lines in $PG(2, q)'$ relative to a regular hyperoval in $PG(2, q)$. The results are summarized at the end of the chapter in Table 6.2. Although the results generally apply to subsets of points and lines with respect to an arbitrary regular hyperoval, for the sake of construction, we can always choose the hyperoval to be the $H_{\mathcal{C}}$ just defined. When analyzing the codes constructed from these subsets, we will often rely on the graphical interpretations. In particular, since a codeword in a classical code must be orthogonal to every row of a parity check matrix, we can graphically interpret a codeword to be a collection of points, S, in the plane having the property that each line corresponding to a row in the parity check matrix contains an even number of points in S.

Table 6.1: Number of points and lines with respect to the hyperoval $H_{\mathcal{C}}$ and non-hyperoval points in $PG(2,q)$.

# of points in $H_{\mathcal{C}}$	# of lines secant to $H_{\mathcal{C}}$	# of lines skew to $H_{\mathcal{C}}$	# of secant lines intersecting a non-hyperoval point	# of skew lines intersecting a non-hyperoval point
$q+2$	$\frac{q^2+3q+2}{2}$	$\frac{q^2-q}{2}$	$\frac{q+2}{2}$	$\frac{q}{2}$

6.1 Skew Lines and Non-Hyperoval Points, C_{sk} and $H(C_{sk})$

Notice that by definition, skew lines never intersect the hyperoval. Suppose we construct a classical parity check matrix whose rows are the subset of rows in M'_π corresponding to skew lines. Then graphically, a codeword for this code is a collection of points S such that each skew line intersects S in an even number of places. Notice that such an example is found by letting S simply be a hyperoval point. Then every skew line intersects S nowhere. Since S contains only one point, it corresponds to a codeword of weight one, and hence this code has minimum weight one. Thus, in general, the parity check matrices constructed from skew lines do not define good codes. However, the removal of hyperoval points may resolve this problem. We will now consider the classical codes whose parity check matrices are created from skew lines in $PG(2,q)'$ with the hyperoval points removed, denoting the code by C_{sk}, and the corresponding parity check by $H(C_{sk})$. Note that $H(C_{sk})$ is simply the incidence matrix M'_π with columns corresponding to hyperoval points and rows corresponding to secant lines removed.

As mentioned earlier, the projective plane $\pi = PG(2,q)$ contains q^2+q+1 points, and the extended projective plane π' contains one more additional point. Since we remove the $q+2$ columns of M'_π corresponding to the hyperoval points, it follows that the length of the rows in $H(C_{sk})$, and thus the length of the codewords, is q^2. Additionally, it is easy to see that, since each skew line contains the u-point, along with $q+1$ other non-hyperoval points, the weight of each row in $H(C_{sk})$ is exactly $q+2$. The number of rows in $H(C_{sk})$ is

$(q^2 - q)/2$, being the number of skew lines in the plane. We would like to put a bound on the minimum distance.

Proposition 6.1. *Suppose S is a collection of points in π' that make up a codeword in C_{sk}. If S does not contain the u-point, then the codeword has a weight of at least $\frac{q}{2} + 1$.*

Proof. S can be viewed as a non-empty collection of non-hyperoval points in $PG(2, q)$. Any skew line that intersects S must do so in an even number of points. Let p be a point in S. We know that it is intersected by $\frac{q}{2}$ skew lines. Then for each of these skew lines, S must contain at least one other point through which the line intersects. The result follows. □

Notice that when the u-point is not in S, we can solve our problem by simply considering points and lines in the standard projective plane. Many of the results in this report are obtained by making an assumption on the the u-point, and then studying how the codewords act in relation to the standard projective plane, as opposed to the extended projective plane. The next result is another such example.

Proposition 6.2. *Suppose S is a collection of points in π' that make up a codeword in C_{sk}. If S contains the u-point, then the codeword has a weight of at least q.*

Proof. Each skew line must intersect S in an even number of points. We assume that h is in S. We can reduce this problem to finding a minimal set of points in $PG(2, q)$ through which each skew line intersects an odd number of times, namely, at least once. But since any two lines in $PG(2, q)$ intersect at exactly one point, we obtain a minimum when we choose S to be a line. Since hyperoval points are removed from the plane, secant lines have the smallest number of points (namely, q points), and they intersect every skew line exactly once in points other than hyperoval points. Thus, when S is a secant line, it will have weight q, and will be a minimum weight among codewords containing the u-point. □

This leads us to the following result about the minimum weight of the code C_{sk}.

Corollary 6.3. *The minimum weight, d_{sk}, of C_{sk} is bounded by $\frac{q}{2} + 1 \leq d_{sk} \leq q$.*

Proof. This follows immediately from Propositions 6.1 and 6.2. Since we know codewords of weight q exist, this value acts as an upper bound. □

In order to fully characterize the classical code constructed from $H(C_{sk})$, we need to determine its dimension. Let $H(C_{sk})\backslash u$ be the matrix formed by removing the u-column from $H(C_{sk})$. Bounds on $\dim(H(C_{sk})\backslash u)$ were found in [22], and we include their results as a proposition here.

Proposition 6.4. *Let $H(C_{sk})\backslash u$ be defined as above. Then $3^s - 2^s \leq \dim(H(C_{sk})\backslash u) \leq 3^s + 1$.*

Proof. We know that $\dim(M_\pi) = 3^s + 1$ from [19]. In [22] it was shown that removing all the rows in M_π corresponding to lines secant to the conic $\mathcal{C}$ will not affect the dimension. Since skew lines never intersect hyperoval points, removing the columns corresponding to hyperoval points does not affect the dimension. Removing lines tangent to $\mathcal{C}$ may possibly decrease the dimension, giving us the resulting bounds. □

Corollary 6.5. *The dimension of $H(C_{sk})$ is bounded by $3^s - 2^s \leq \dim(H(C_{sk})) \leq 3^s + 1$.*

Proof. This follows immediately from Lemma 5.1 and Proposition 6.4. □

6.2 Secant Lines and All Points, C_{seA} and $H(C_{seA})$

We showed that, in order to obtain good minimum distance codes whose parity checks are created from skew lines, we needed to remove hyperoval points. Since every secant line intersects the hyperoval, this requirement is not necessary. We will now describe the classical codes whose parity checks are the incidence matrices constructed from all points on the extended projective plane π' and lines secant to the hyperoval, denoting the classical code by C_{seA} and the corresonding parity check matrix by $H(C_{seA})$.

Observe that the codewords here will have length $q^2+q+2 = 4^s+2^s+2$, while the weight of each parity check row is $q+1$. $H(C_{seA})$ will have $(q^2+3q+2)/2$ rows, corresponding

to the number of secant lines in the projective plane. We will now discuss bounds of the minimum weight of C_{seA}.

Proposition 6.6. *Suppose S is a collection of points in π' that make up a codeword in C_{seA}. If S does not contain the u-point, then the codeword has a weight of at least $\frac{q}{2}+2$.*

Proof. We can view S as a nonempty collection of points in $PG(2,q)$. Any secant line that intersects S must do so in an even number of points. Let p be a point in S. We know that it is intersected by $(q+2)/2$ secant lines. Thus S must contain at least one more additional point for each of those lines. □

Proposition 6.7. *Suppose S is a collection of points in π' that make up a codeword in C_{seA}. If S contains the u-point, then the codeword has a weight of at least $q+2$.*

Proof. Since each secant line must intersect S in an even number of points, and one of those points is the u-point, then the remaining points in S must be points in $PG(2,q)$ intersected by each secant line in $PG(2,q)$ an odd number of times, in particular at least once. Since each line in $PG(2,q)$ intersects all other lines exactly once, we obtain a minimum when we choose the remaining points to be a collection of $q+1$ points that make up a line in $PG(2,q)$. □

Thus, we obtain the following result about the overall minimum weight of C_{seA}.

Corollary 6.8. *The minimum weight d_{seA} of the classical code C_{seA} is bounded by $2^{s-1}+2 \leq d_{seA} \leq 2^s+2$.*

The following result allows us to completely characterize the code.

Proposition 6.9. *The dimension of the parity check matrix $H(C_{seA})$ for the classical code C_{seA} is given by $\dim(H(C_{seA})) = 3^s+1$.*

Proof. Let M_π be the incidence matrix for the projective plane $\pi = PG(2,q)$, again having known rank of 3^s+1 from [19]. It was shown in [22] that the rows corresponding to skew

lines can be removed without affecting the dimension of this matrix. Denote by $H(C_{seA})\backslash u$ the resulting incidence matrix. Since each row of $H(C_{seA})\backslash u$ has odd weight, it follows from Lemma 5.1 that we can concatenate the u-column without changing the dimension (rank). The resulting matrix is precisely $H(C_{seA})$, and hence $\dim(H(C_{seA})) = 3^s + 1$. □

6.3 Secant Lines and Non-Hyperoval Points, C_{se} and $H(C_{se})$

As discussed in the previous section, it is not necessary to remove hyperoval points from incidence matrices constructed from secant lines in order to construct classical codes with good minimum distance. In this section, we will remove the hyperoval points nonetheless, as this will be necessary when constructing the parity check matrix for an asymmetric quantum CSS code constructed from both subsets of lines. Namely, since the length of the parity check matrix created from secant lines must be the same as that created from the skew lines, we must eliminate $q+2$ columns from the incidence matrix created from secant lines and points in π'. In order to make sure we maintain orthogonality between the two matrices, we choose to eliminate the columns corresponding to hyperoval points from this incidence structure as well. We denote the corresponding parity check matrix by $H(C_{se})$ and the code space by C_{se}. Notice, however, that by removing hyperoval points, secant lines will not always overlap twice. What this means in terms of the Tanner graph is a reduction of four-cycles, potentially giving this code a slight advantage over C_{sk} and C_{seA}.

Notice that by removing the columns corresponding to hyperoval points from the incidence matrix, the length of the rows of $H(C_{se})$, and therefore the length of the codewords in C_{se}, is q^2. Moreover, the weight of each row in the parity check is q since each secant line in the extended projective plane intersects q non-hyperoval points. $H(C_{se})$ will have $(q^2+3q+2)/2$ rows, corresponding to the number of secant lines in the projective plane. We will now discuss bounds on the minimum weight of C_{se}.

Proposition 6.10. *Suppose S is a collection of points in π' that make up a codeword in C_{se}. If S does not contain the u-point, then the codeword has a weight of at least $\frac{q}{2}+2$.*

Proof. We can view S as a non-empty collection of non-hyperoval points in $PG(2,q)$. Any secant line that intersects S must do so in an even number of points. Let p be a point in S. We know that it is intersected by $(q+2)/2$ secant lines. Then for each of these secant lines, S must contain at least one other point through which the line intersects. The result follows. □

Proposition 6.11. *Suppose S is a collection of points in π' that make up a codeword in C_{se}. If S contains the u-point, then the codeword has a weight of at least $q+2$.*

Proof. The proof is similar to that of Proposition 6.2. We can reduce this problem to finding a minimal set of points in $PG(2,q)$ through which each secant line intersects an odd number of times, namely, at least once. But since any two lines in $PG(2,q)$ intersect at exactly one point, we obtain a minimum when we choose S to be a line. Since hyperoval points are removed from the plane, there exist pairs of secant lines that do not have points in common. The only lines in $PG(2,q)$ that intersect every secant line exactly once at non-hyperoval points are skew lines, having weight $q+2$ in the extended projective plane. Thus, when S corresponds to a skew line, it will have weight $q+2$, and the codeword will have minimum weight among codewords containing the u-point. □

This leads us to the following result about the minimum weight of the code C_{se}.

Corollary 6.12. *The minimum weight, d_{se}, of C_{se} is bounded by $\frac{q}{2}+2 \leq d_{se} \leq q+2$.*

Proof. The proof follows immediately from Propositions 6.10 and 6.11. Again, since we know codewords of weight $q+2$ exist, this value acts as an upper bound. □

We now want to determine the dimension of $H(C_{se})$.

Proposition 6.13. $\dim(H(C_{se})) = 3^s + 1$.

Proof. We already know $\dim(M_\pi) = 3^s+1$ for $\pi = PG(2,2^s)$. It was shown in [22] that columns corresponding to hyperoval points and rows corresponding to skew lines can be removed from M_π without affecting the dimension. This new matrix is simply $H(C_{se})$

Table 6.2: Classical codes and their corresponding parameters n, k, and d. Recall that the dimension k of a classical code C with parity check matrix $H(C)$ is given by $k = n - \dim(H(C))$.

	Codeword Length n	Code Dimension k	Code Minimum Distance d
C_{sk}	4^s	$4^s - 3^s - 1 \leq k \leq 4^s - 3^s + 2^s$	$2^{s-1} + 1 \leq d \leq 2^s$
C_{seA}	$4^s + 2^s + 2$	$4^s - 3^s + 2^s + 1$	$2^{s-1} + 2 \leq d \leq 2^s + 2$
C_{se}	4^s	$4^s - 3^s + 2^s + 1$	$2^{s-1} + 2 \leq d \leq 2^s + 2$

with the u-column removed, denoted $H(C_{se})\backslash u$. Note that each row of $H(C_{se})\backslash u$ has odd weight, and so by Lemma 5.1, we can add the u-column without affecting the rank. Hence, $\dim(H(C_{se})) = \dim(H(C_{se})\backslash u) = 3^s + 1$. □

Chapter 7: Quantum LDPC Codes from Point-Line Subsets

We are now ready to discuss the results as they pertain to quantum low-density parity check (QLDPC) codes. We use the incidence matrices of subsets of points and lines in the extended projective plane, and results about the corresponding classical codes, to construct both symmetric and asymmetric QLDPC codes.

7.1 Asymmetric QLDPC Codes

We first analyze the asymmetric QLDPC code C_{asym} with parity check matrix $H(C_{asym})$ of the following form.

$$H(C_{asym}) = \begin{bmatrix} H(C_{sk}) & 0 \\ 0 & H(C_{se}) \end{bmatrix} \tag{7.1}$$

Recall that $H(C_{sk})$ and $H(C_{se})$ are orthogonal by construction, and both have the same block length. Thus $H(C_{asym})$ is a parity check matrix for a quantum CSS code. Since $H(C_{sk})$ and $H(C_{se})$ are both parity check matrices for classical LDPC codes (i.e. they are sparse and have few four-cycles), it follows that $H(C_{asym})$ is in fact a parity check matrix for a QLDPC code. $H(C_{asym})$ has q^2+q+1 rows and $2q^2$ columns. The Tanner graph of this code will in turn have q^2+q+1 check nodes, and q^2 bit nodes. In terms of stabilizer codes, this means that there will be q^2+q+1 stabilizers of length q^2. We conclude this section with the following theorem.

Theorem 7.1. Given a finite projective plane $PG(2,2^s)$, for some positive integer s, the matrix $H(C_{asym})$ in (7.1) is a parity check matrix for a $[[4^s, K, D]]$ QLDPC code C_{asym},

where the dimension K is bounded by $4^s - 2\cdot 3^s + 2 \leq K \leq 4^s - 2\cdot 3^s + 2^s - 1$, and the overall minimum distance D is bounded by $2^{s-1} + 1 \leq D$. C_{asym} will have $4^s + 2^s + 1$ stabilizers.

Proof. We already showed that $H(C_{asym})$ is a parity check matrix for a QLDPC code C_{asym}. The number of stabilizers corresponds to the number of lines in $PG(2, 2^s)$. The length is determined by observing that $H(C_{sk})$ and $H(C_{se})$ are parity checks for classical codes of length 4^s. Since $\dim(C_{sk}) = k_{sk}$ is bounded by $4^s - 3^s - 1 \leq k_{sk} \leq 4^s - 3^s + 2^s$ and $\dim(C_{se}) = k_{se} = 4^s - 3^s + 2^s + 1$ (see Table 6.2), then we obtain the dimension by noting that $\dim(C_{asym}) = k_{sk} + k_{se} - n$. The minimum distance D is bounded below by $\min(d_{sk}, d_{se})$, where d_{sk} and d_{se} are the minimum distances of the classical codes C_{sk} and C_{se}, respectively. Using Corollaries 6.3 and 6.12, the result follows. □

While the bound on the dimension may be coarse, it is important to observe that the rate of these codes nevertheless increases rapidly with the code length n. The code will have only $n + \sqrt{n} + 1$ parity checks, and very few four-cycles.

7.2 Symmetric QLDPC Codes from Skew Lines

Here we will analyze QLDPC codes C_{symSK} whose parity check matrices $H(C_{symSK})$ have the following form.

$$H(C_{symSK}) = \begin{bmatrix} H(C_{sk}) & 0 \\ 0 & H(C_{sk}) \end{bmatrix} \tag{7.2}$$

Since $H(C_{sk})$ is a self-dual parity check (and hence satisfies the twisted inner product requirement) for a classical LDPC code, it follows that $H(C_{symSK})$ is, in fact, a parity check for a QLDPC code. This parity check matrix has $2\left(\frac{q^2-q}{2}\right)$ rows and $2q^2$ columns, and hence the Tanner graph of this code will have $q^2 - q$ check nodes and q^2 bit nodes. In terms of stabilizer codes, this means that there will be $q^2 - q$ stabilizers of length q^2. We conclude this section with the following theorem.

Theorem 7.2. Given a finite projective plane $PG(2, 2^s)$, for some positive integer s, the matrix $H(C_{symSK})$ in (7.2) is a parity check for a $[[4^s, K, D]]$ QLDPC code C_{symSK}, where the dimension K is bounded by $4^s - 2 \cdot 3^s - 2 \leq K \leq 4^s - 2 \cdot 3^s + 2^{s+1}$, and the minimum distance D is bounded by $2^{s-1} + 1 \leq D$.

Proof. We already showed that $H(C_{symSK})$ is in fact a well-defined QLDPC code. Clearly, $n = q^2 = 4^s$. The dimension is obtained by observing that $K = 2k_{sk} - n$, where $k_{sk} = \dim(C_{sk})$. The minimum distance D is bounded below by the minimum distance of the classical code C_{sk}, which we obtain from Corollary 6.3. □

Although the bound on the dimension of these codes is not as tight as that of the asymmetric codes described in Section 7.1, these codes nevertheless have a fast rate that increases with the length n. The bound on the minimum distance is the same as for the codes in Section 7.1, showing that these, too, describe fast-rate QLDPC codes with good minimum distance. The codes will have $n - \sqrt{n}$ parity checks and few four-cycles.

7.3 Symmetric QLDPC Codes from Secant Lines

Here we will analyze QLDPC codes C_{symSE} whose parity check matrices $H(C_{symSE})$ have the following form.

$$H(C_{symSE}) = \begin{bmatrix} H(C_{seA}) & 0 \\ 0 & H(C_{seA}) \end{bmatrix} \tag{7.3}$$

Again, we observe that the twisted inner product is statisfied by the fact that $H(C_{seA})$ is self-orthogonal by construction. Since $H(C_{seA})$ is a parity check for a classical LDPC code, it follows that $H(C_{symSE})$ truly is a parity check matrix for a QLDPC code. This matrix has $2\left(\frac{q^2+3q+2}{2}\right)$ rows and $2(q^2 + q + 2)$ columns, and hence the Tanner graph of this code will have $q^2 + 3q + 2$ check nodes and $q^2 + q + 2$ bit nodes. In terms of stabilizer codes,

this means that there will be $q^2 + 3q + 2$ stabilizers of length $q^2 + q + 2$. We conclude this section with the following theorem.

Theorem 7.3. Given a finite projective plane $PG(2, 2^s)$, for some positive integer s, the matrix $H(C_{symSE})$ in (7.3) is a parity check matrix for a $[[4^s + 2^s + 2, 4^s - 2 \cdot 3^s + 2^s, D]]$ QLDPC code C_{symSE}, where the minimum distance D is bounded by $2^{s-1} + 2 \leq D$.

Proof. The proof is similar to that of Theorems 7.1 and 7.2, and relies on the results found in Section 6.2. □

While n is slightly larger here than for C_{asym} and C_{symSK}, its exact dimension may make this a slightly more preferable code, especially given that the bound on the minimum distance is roughly the same as those of the other two classes.

Note that we did not build the parity check matrix for the above QLDPC code from secant lines and non-hyperoval points $H(C_{se})$. This is because the removal of hyperoval points destroys the self-orthogonality of $H(C_{se})$, and so the matrix

$$\begin{bmatrix} H(C_{se}) & 0 \\ 0 & H(C_{se}) \end{bmatrix}$$

does not correspond to a CSS code.

Chapter 8: Conclusion

Table 8.1 gives a summary of the results for each of the QLDPC codes discussed in this report.

Table 8.1: QLDPC Code Parameters for Parity Checks Constructed from $PG(2, 2^s)$

Code	**Length**	**Dimension**	**Minimum Distance (Lower Bound)**	**Number of Stabilizers**
C_π	$4^s + 2^s + 2$	$4^s - 2 \cdot 3^s + 2^s$	$2^s + 2$	$2^{2s+1} + 2^{s+1} + 2$
C_{asym}	4^s	$4^s - 2 \cdot 3^s + 2 \leq K \leq 4^s - 2 \cdot 3^s + 2^s - 1$	$2^{s-1} + 1$	$4^s + 2^s + 1$
C_{symSK}	4^s	$4^s - 2 \cdot 3^s - 2 \leq K \leq 4^s - 2 \cdot 3^s + 2^{s+1}$	$2^{s-1} + 1$	$4^s - 2^s$
C_{symSE}	$4^s + 2^s + 2$	$4^s - 2 \cdot 3^s + 2^s$	$2^{s-1} + 2$	$4^s + 3 \cdot 2^s + 2$

While many of the parameters are not exact for these codes, the bounds nevertheless indicate that each of these codes are at least comparable to most quantum LDPC codes in the literature. In fact, as previously mentioned, many of these parameters are completely unknown for the other quantum LDPC codes. Moreover, although many of the bounds given here are relatively weak, simulations of the classical codes seem to indicate that the actual values for most of these parameters are in fact their upper bounds [22].

Note that the code C_π has a minimum distance that increases faster than $\sqrt{n}$, and both C_{symSE} and C_{asym} have the potential to do the same. This is particularly notable, given that this appears to be faster than the increase of any known minimum distances of

any other quantum LDPC codes. C_π and C_{symSE} have the same rate, and potentially the same minimum distance. The distinct difference between these two codes is in the number of stabilizers in their parity checks, with C_{symSE} having significantly fewer (in particular, $4^s - 2^s$ fewer). This indicates that C_{symSE} requires far fewer check nodes for decoding, and could potentially have the same error correction performance as that of C_π. Simulations will be needed to verify this.

The distinct advantage that C_{symSK} has over the other codes is the small number of stabilizers (or equivalently, the small number of check nodes in the corresponding Tanner graph), indicating that this code may potentially be easier to physically implement. Another potential advantage is that if the dimension of this code is in fact its upper bound, then C_{symSK} will have a rate faster than all the others. Additionally, if the minimum weight of the classical code C_{sk} is in fact the upper bound of 2^s from Corollary 6.3, then the minimum distance would also be bounded below by exactly $\sqrt{n}$.

While further research is necessary to determine exact values of minimum distance and dimension of most of these codes, it is nevertheless established that the projective plane is a very useful tool in the construction of quantum low-density parity check codes. Similar techniques can also be used to construct QLDPC codes from $PG(m, p^s)$ for $m > 2$ and/or p an odd prime. Note that in the case of $PG(2, p^s)$ where p is an odd prime, the u-column is not necessarily linearly dependent on the columns of the corresponding incidence matrix, making it much more difficult to determine dimensions of corresponding QLDPC codes. However, this is resolved if in addition to concatenating the u-column to the incidence matrix of the projective plane, you also concatenate an identity matrix to it. This would change the parameters in a known manner. Moreover, when p is an odd prime, we cannot in general extend a conic to a regular hyperoval, and thus a different method of splitting the incidence matrix of $PG(2, p^s)$ must be used.

Bibliography

Bibliography

[1] MacKay, Mitchison, and McFadden, "Sparse-graph codes for quantum error correction," *IEEE Transactions on Information Theory*, vol. 50, no. 10, 2004.

[2] Postol, "A proposed quantum low density parity check code," *arXiv:quant-ph/0108131*, 2001.

[3] J.-P. Tillich and G. Zémor, "Quantum ldpc codes with positive rate and minimum distance proportional to n 1/2," in *Proceedings of the 2009 IEEE international conference on Symposium on Information Theory - Volume 2*, ser. ISIT'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 799–803. [Online]. Available: http://dl.acm.org/citation.cfm?id=1701275.1701299

[4] S. Aly, "A class of quantum ldpc codes constructed from finite geometries," in *Proc. 2008 IEEE Global Communication*, ser. Globecom '08, New Orleans, LA, USA, 2008.

[5] V. Pless, *Introduction to the Theory of Error-Correcting Codes*, 3rd ed. Wiley-Interscience Series in Discrete Mathematics, 1998.

[6] R. G. Gallager, "Low density parity check codes," *Transactions of the IRE Professional Group on Information Theory*, vol. IT-8, pp. 21–28, January 1962.

[7] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379 – 423, 623 – 656, 1948.

[8] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge, England: Cambridge University Press, 2000.

[9] P. W. Shor, "Polynomial-time algorithm for prime factorization and discrete logarithms on a quantum computer," 1996.

[10] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, Oct. 1982. [Online]. Available: http://dx.doi.org/10.1038/299802a0

[11] D. Dieks, "Communication by EPR devices," *Physics Letters A*, vol. 92, no. 6, pp. 271–272, Nov. 1982. [Online]. Available: http://dx.doi.org/10.1016/0375-9601(82)90084-6

[12] D. Gottesman, "Class of quantum error-correcting codes saturating the quantum hamming bound," *Phys. Rev. A*, vol. 54, pp. 1862–1868, 1996.

[13] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum error correction and orthogonal geometry," *Phys. Rev. Lett.*, vol. 78, no. 3, 1997.

[14] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *arXiv:quant-ph/9512032*, 1996.

[15] A. M. Steane, "Multiple particle interference and quantum error correction," *Proc. R. Soc. Lond. A*, vol. 452, no. 2551, 1996.

[16] ——, "Error correcting codes in quantum theory," *Phys. Rev. Lett.*, vol. 77, pp. 793 – 797, 1996.

[17] J. Singer, "A theorem in finite projective geometry and some applications to number theory," *Trans. Am. Math. Soc.*, vol. 43, pp. 377 – 385, 1938.

[18] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low density parity check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2711–2736, 2001.

[19] K. J. C. Smith, "On the p-rank if the incidence matrix of points and hyperplanes in a finite projective geometry," *J. Comb. Theory*, vol. 7, pp. 122 – 129, 1969.

[20] P. J. Cameron, *Combinatorics: Topics, Techniques, Algorithms.* Cambridge University Press, 1994.

[21] Droms, Meyer, and K. E. Mellinger, "Ldpc codes generated by conics in the classical projective plane," *Designs, Codes, and Cryptography*, vol. 40, no. 3, 2006.

[22] Castleberry, K. Hunsberger, and K. E. Mellinger, "Ldpc codes arising from hyperovals," *Bull. Inst. Combin. Appl.*, vol. 58, pp. 59 – 72, 2010.

Curriculum Vitae

Jacob M. Farinholt received his Bachelor of Science in Mathematics from the University of Mary Washington in 2009. He currently works for the U.S. Navy as a Scientist. He received his Master of Science in Mathematics from George Mason University in 2012.