QIP Seminar: Quantum Error Correcting and Q-LDPC

Eshed Ram

1 Introduction

Since the dawn of the digital-information age, error-correcting codes played a crucial rule in enabling reliable data transfer (or storage). The most basic model is a follows: Alice wishes to send information

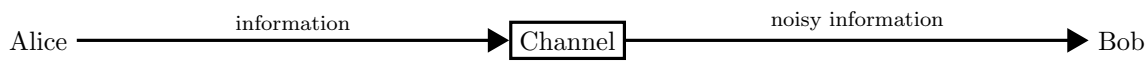

Figure 1: A noisy channel.

(classic or quantum) to Bob, but the *channel* corrupts the bits/qubits so that the transmission may fail (note that the channel is probabilistic and not viscous). To overcome the noise inserted by the channel, an error-correcting code is used. Alice *encodes* the information into codewords from a

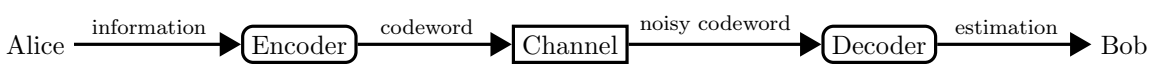

Figure 2: An error-correcting scheme.

codebook agreed upon in advance with the Bob. This is a one-to-one mapping that adds redundancy to information. Consequently, Bob *decodes* the noisy signal and uses the redundancy to estimate the sent codeword.

Example 1 *Consider a classical bit sent from Alice to Bob. Assume the channel flips the bit with probability* $p = 10^{-1}$. *If Alice would replicate the bit* 3 *times, then Bob may be able to reconstruct it by a majority vote. The error probability using this coding scheme is the probability that the channel flipped at least two bits out of the three sent, i.e.,* $P_e = \binom{3}{2}p^2(1-p) + p^3 = 2.8 \cdot 10^{-2}$.

It is quite clear that as Alice adds more redundancy, then Bob is more likely to successfully reconstruct her message. However, adding redundancy is costly (more bits/qubits needed). Coding theory and information theory deal with this conflict by stating what are the fundamental limits of the scheme (i.e., the minimal redundancy needed), and finding error-correcting codes that achieve these limits with low-complexity encoding and decoding algorithms. Low-density parity-check (LDPC) codes are state-of-the-art classical error-correcting codes that approach the fundamental limits with very low-complexity decoding algorithms.

When discussing quantum information, few difficulties arise. First, one cannot implement a repetition code by duplication due to the no-cloning principle. Second, quantum errors – as quantum states – are continuous, so determining which error occurred could require infinite precision. Last but not least, the decoder should not measure the received state to prevent it from collapsing, so its decoding procedure cannot depend on this measurement (unlike the classical case). Fortunately,

Quantum error-correcting codes (QECC) overcome these problems. For example, in the introductory course, we learned about the *three-qubits bit-flip code* and *Shor's code*. The former can correct a single bit flip (similar to the classical 3x-repetition code), and the latter can correct a single bit flip and a single phase flip (and both!) by encoding one qubit into nine qubits.

This seminar will discuss quantum error correcting in more details with the goal of presenting stabilizer codes and the quantum counterpart of LDPC codes (Q-LDPC). Toward this goal, concepts from classical codes are reviewed as well as basic concepts from quantum error correcting. Sections 2-3 can be found with more details in [1], and Section 4 is based on [2] and references therein.

2 Preliminaries

2.1 Linear Codes

Mathematically speaking, a classic error-correcting code $\mathcal{C}$ is a set of M words, each contains letters from some alphabet F. Usually, the alphabet F is a field (additive group, multiplicative group, and distributive) and the words (a.k.a codewords) are of the same length n, i.e., $\mathcal{C} \subseteq F^n$. We will consider the Binary field where all operations are performed modulo 2. Linear codes are a special case of error-correcting codes, although used in the vast majority of problems.

Definition 1 *An error correcting code over F is said to be a linear code if it is a linear sub-space of F^n. In this case, we mark $k \triangleq \dim(\mathcal{C})$, and say that $\mathcal{C}$ is an $[n,k]$ code.*

Example 2 *The repetition code is a $[n,1]$ linear code.*

As a linear sub-space, there exist a basis $\{\underline{e}_1,\ \underline{e}_2,\ \ldots,\ \underline{e}_k\}$ that spans $\mathcal{C}$. In other words, each codeword $\underline{c} \in \mathcal{C}$ can be represented as a linear combination $\underline{c} = \sum_{i=1}^{k} u_i \underline{e}_i$, where $u_i \in F$ are scalars. Locating the basis $\{\underline{e}_i\}$ into k rows yields a matrix G called *the generator matrix*, such that $\underline{c} = \underline{u}G$. This last equation describes *encoding* – the process of adding redundancy to information. In fact, G specifies the code by

$$\mathcal{C} = \{\underline{u}G \colon \underline{u} \in F^k\}.$$

When talking about error-correction, an alternative (yet equivalent) specification of the code should be introduced.

Definition 2 *Let $\mathcal{C}$ be an $[n,k]$ code over F. A parity-check matrix of $\mathcal{C}$ is an $(n-k) \times n$ matrix H over F such that*

$$\mathcal{C} = \ker(H) \triangleq \{\underline{c} \in F^n \colon H\underline{c}^T = 0\}\ .$$

Assume $\underline{c} \in \mathcal{C}$ is sent over a channel that may flip some bits, and let $\underline{y}$ be the corrupted word. We can write $\underline{y} = \underline{c} + \underline{e}$, where the vector $\underline{e}$ indicates the bit-flip positions and the addition is done modulo 2. Since $H\underline{c}^T = 0$, then by calculating $\underline{s}^T = H\underline{y}^T = H\underline{e}^T$, we can: 1) detect an error if $\underline{s} \neq 0$; 2) learn something about the error. $\underline{s}$ is called the *syndrome*, and it has a quantum analogue (later). Each code can be characterized by a parameter t that indicates the number of bit flips tolerable using this code. Ideally, this is a large number, but enlarging it costs with the need of more redundancy.

Example 3 *The parity-check matrix of the $[n,1]$ repetition code is given by*

$$H = \left(\begin{array}{c|c} & 1 \\ & 1 \\ I_{n-k} & \vdots \\ & 1 \\ & 1 \end{array}\right)$$

The repetition code can correct upto $t = \lfloor \frac{n-1}{2} \rfloor$ errors.

Example 4 *The Hamming code of order $m \geq 3$ is an $[n = 2^m - 1, k = 2^m - 1 - m]$ code that has a parity-check matrix H with the following structure: the columns of H are all of the non-zero binary vector of length m. For example, for $m = 3$ we get*

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

The Hamming code, of every order m, can correct up to $t = 1$ errors.

The last linear-codes topic needed to be covered are *dual* codes. If $\mathcal{C}$ is an $[n, k]$ code specified by a generator matrix G and a parity-check matrix H, then its dual is a $[n, n-k]$ code marked by $\mathcal{C}^\perp$ specified by a generator matrix H and a parity-check matrix G. Equivalently, the dual code $\mathcal{C}^\perp$ contains all words that are "orthogonal" to the words in $\mathcal{C}$,

$$\mathcal{C}^\perp = \{\underline{x} \colon \underline{x} \cdot \underline{c}^T = 0, \ \forall \underline{c} \in \mathcal{C}\}. \tag{2}$$

From coset arguments we have

$$\sum_{\underline{c} \in \mathcal{C}} (-1)^{\underline{c} \cdot \underline{x}^T} = \begin{cases} |\mathcal{C}| & \underline{x} \in \mathcal{C}^\perp \\ 0 & \underline{x} \notin \mathcal{C}^\perp \end{cases}. \tag{3}$$

Example 5 *The dual of the Hamming code of order $m = 3$ is specified by a generator matrix*

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

The dual Hamming code is weekly self dual, *i.e.,* $\mathcal{C}_{Ham}^\perp \subseteq \left(\mathcal{C}_{Ham}^\perp\right)^\perp = \mathcal{C}_{Ham}$.

2.2 Quantum Error Correcting: Calderbank-Shor-Steane Codes

In the introductory course we saw that every error on a qubit is one of four options: 1) the no-error error; 2) the bit-flip error; 3)the phase-flip error; 4) the bit-and-phase-flip error. If $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ is a qubit state then these errors corrupt it as follows

$$\begin{aligned} |\psi\rangle &\overset{1)}{\to} I |\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \\ |\psi\rangle &\overset{2)}{\to} X |\psi\rangle = \beta |0\rangle + \alpha |1\rangle, \\ |\psi\rangle &\overset{3)}{\to} Z |\psi\rangle = \alpha |0\rangle - \beta |1\rangle, \\ |\psi\rangle &\overset{4)}{\to} ZX |\psi\rangle = \beta |0\rangle - \alpha |1\rangle. \end{aligned}$$

In addition, we can encode qubits by $|0\rangle \to |0_L\rangle$. $|1\rangle \to |1_L\rangle$, where $|0_L\rangle$, $|1_L\rangle$ represent the logical '1' and '0' states, respectively. Then, by linearity $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ is encoded to $\alpha |0_L\rangle + \beta |0_L\rangle$.

Example 6

1. *Single bit-flip code: $|0_L\rangle = |000\rangle$, $|1_L\rangle = |111\rangle$. Can correct a single bit-flip error by measuring the observables $Z \otimes Z \otimes I$ and $I \otimes Z \otimes Z$. Since $Z \otimes Z \otimes I = (|00\rangle\langle 00| + |11\rangle\langle 11|) - (|10\rangle\langle 10| + |01\rangle\langle 01|) \otimes I$, then measuring it will give $+1$ if qubits 1 and 2 are the same, and -1 otherwise (similarly for $I \otimes Z \otimes Z$ and qubits 2 and 3).*

2. *Single phase-flip code:* $|0_L\rangle = |+++\rangle$, $|1_L\rangle = |---\rangle$. *Can correct a single phase-flip error by measuring the observables* $X \otimes X \otimes I$ *and* $I \otimes X \otimes X$.

3. *Shor's code:*

$$|0_L\rangle = \tfrac{1}{2\sqrt{2}} \left(|000\rangle + |111\rangle\right) \left(|000\rangle + |111\rangle\right) \left(|000\rangle + |111\rangle\right),$$
$$|1_L\rangle = \tfrac{1}{2\sqrt{2}} \left(|000\rangle - |111\rangle\right) \left(|000\rangle - |111\rangle\right) \left(|000\rangle - |111\rangle\right).$$

Can correct a single arbitrary error.

2.2.1 Colderbank-Shor-Steane Codes

One nice example of a family of QECC are *Colderbank-Shor-Steane* (CSS) codes named after their inventors. These codes use the properties of classical linear codes to implement quantum codes. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be an $[n, k_1]$ and an $[n, k_2]$ linear codes, respectively, such that $\mathcal{C}_2 \subset \mathcal{C}_1$, and both $\mathcal{C}_1$ and $\mathcal{C}_2^\perp$ can correct up to t errors. For any $\underline{x} \in \mathcal{C}_1$, define the next quantum state

$$|\underline{x} + \mathcal{C}_2\rangle = \frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{\underline{y} \in \mathcal{C}_2} \left|\underline{x} + \underline{y}\right\rangle. \tag{4}$$

Then the following properties hold:

1. If $\underline{x}$, $\underline{x}' \in \mathcal{C}_2$, then $|\underline{x} + \mathcal{C}_2\rangle = |\underline{x}' + \mathcal{C}_2\rangle$.
2. More generally, if $\underline{x}$, $\underline{x}'$ belong to the same *coset* of $\mathcal{C}_2$, then $|\underline{x} + \mathcal{C}_2\rangle = |\underline{x}' + \mathcal{C}_2\rangle$.
3. If $\underline{x}$, $\underline{x}'$ **do not** belong to the same *coset* of $\mathcal{C}_2$, then for every $\underline{y}$, $\underline{y}' \in \mathcal{C}_2$, $\underline{x} + \underline{y} \neq \underline{x}' + \underline{y}'$. This implies that the quantum states $|\underline{x} + \mathcal{C}_2\rangle$ and $|\underline{x}' + \mathcal{C}_2\rangle$ are orthogonal.

Definition 3 *The quantum code* $CSS(\mathcal{C}_1, \mathcal{C}_2)$ *is defined to be the vector space spanned by the (orthnormal) basis* $\{|\underline{x} + \mathcal{C}_2\rangle\}_{\underline{x} \in \mathcal{C}_1}$.

Proposition 1 *A* $CSS(\mathcal{C}_1, \mathcal{C}_2)$ *constructed by the above procedure can correct up to* t *arbitrary qubit errors.*

Before proving $CSS(\mathcal{C}_1, \mathcal{C}_2)$'s correction capabilities, note how these code's improve the performance over Shor's code. As mentioned, Shor's code can correct up to $t = 1$ arbitrary error using 8 redundant qubits. We now show how $CSS(\mathcal{C}_1, \mathcal{C}_2)$ can correct up to $t = 1$ arbitrary error using only 6 redundant qubits. Let $\mathcal{C}_1$ be the $[n = 7, k = 4]$ Hamming code corresponding to the parity-check matrix in (1), and let $\mathcal{C}_2 = \mathcal{C}_1^\perp$ be its dual. As stated in Examples 4 and 5, it happens to be that $\mathcal{C}_2 \subset \mathcal{C}_1$, and both $\mathcal{C}_1$ and $\mathcal{C}_2^\perp$ can correct up to $t = 1$ errors. In view of Proposition 1, this 7-qubits CSS code can correct up to $t = 1$ arbitrary qubit errors.

Proof 1 (Proof of Proposition 1) *Let* $\underline{e}_X$ *(resp.* $\underline{e}_Z$*) be an* n*-bit vector that describes the locations of the bit-flip (resp. phase-flip) errors. Suppose that the original state was* $|\underline{x} + \mathcal{C}_2\rangle$ *for some* $\underline{x} \in \mathcal{C}_1$*. Then, the corrupted state is given by*

$$\frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{\underline{y} \in \mathcal{C}_2} (-1)^{(\underline{x}+\underline{y}) \cdot \underline{e}_Z^T} \left|\underline{x} + \underline{y} + \underline{e}_X\right\rangle. \tag{5}$$

To correct the bit flips, introduce and ancilla state $n-k_1$ qubits initialized to the zero state, and perform the next reversible computation (syndrome computation)

$$\begin{aligned}\left|\underline{x}+\underline{y}+\underline{e}_X\right\rangle|0\rangle &\to \left|\underline{x}+\underline{y}+\underline{e}_X\right\rangle\left|H_1\left(\underline{x}+\underline{y}+\underline{e}_X\right)^T\right\rangle \\ &= \left|\underline{x}+\underline{y}+\underline{e}_X\right\rangle\left|H_1\underline{e}_X^T\right\rangle\ .\end{aligned} \tag{6}$$

This computation on the state (5) *yields a state*

$$\frac{1}{\sqrt{|\mathcal{C}_2|}}\sum_{\underline{y}\in\mathcal{C}_2}(-1)^{(\underline{x}+\underline{y})\cdot\underline{e}_Z^T}\left|\underline{x}+\underline{y}+\underline{e}_X\right\rangle\left|H_1\underline{e}_X^T\right\rangle. \tag{7}$$

Now, measure the ancilla and discard it. This brings as back to state (5)*, but with information on $H_1\underline{e}_X^T$. Since $\mathcal{C}_1$ can correct up to t (classical) errors, then from $H_1\underline{e}_X^T$ we can infer e_X, and thus by applying CNOT with e_X as the control we can correct all bit-flip errors, given the state*

$$\frac{1}{\sqrt{|\mathcal{C}_2|}}\sum_{\underline{y}\in\mathcal{C}_2}(-1)^{(\underline{x}+\underline{y})\cdot\underline{e}_Z^T}\left|\underline{x}+\underline{y}\right\rangle\ . \tag{8}$$

It is left to show how to correct t phase-flip errors. Using the identity $H^{\otimes n}\left|\underline{x}\right\rangle=\frac{1}{\sqrt{2^n}}\sum_{\underline{z}\in\{0,1\}^n}(-1)^{\underline{x}\cdot\underline{z}^T}\left|\underline{z}\right\rangle$, where H stands for the Hadamard gate, by applying a Hadamard gate on each qubit of (8) *we get*

$$\begin{aligned}\frac{1}{\sqrt{|\mathcal{C}_2|2^n}}\sum_{\underline{z}\in\{0,1\}^n}\sum_{\underline{y}\in\mathcal{C}_2}(-1)^{(\underline{x}+\underline{y})\cdot\left(\underline{e}_Z+\underline{z}\right)^T}\left|\underline{z}\right\rangle &= \frac{1}{\sqrt{|\mathcal{C}_2|2^n}}\sum_{\underline{z}'\in\{0,1\}^n}\sum_{\underline{y}\in\mathcal{C}_2}(-1)^{(\underline{x}+\underline{y})\cdot\left(\underline{z}'\right)^T}\left|\underline{z}'+\underline{e}_Z\right\rangle \\ &= \frac{1}{\sqrt{|\mathcal{C}_2|2^n}}\sum_{\underline{z}'\in\{0,1\}^n}(-1)^{\underline{x}\cdot\underline{z}'^T}\left|\underline{z}'+\underline{e}_Z\right\rangle\sum_{\underline{y}\in\mathcal{C}_2}(-1)^{\underline{y}\cdot\underline{z}'^T} \\ &= \frac{|\mathcal{C}_2|}{\sqrt{|\mathcal{C}_2|2^n}}\sum_{\underline{z}'\mathcal{C}_2^\perp}(-1)^{\underline{x}\cdot\underline{z}'^T}\left|\underline{z}'+\underline{e}_Z\right\rangle\ ,\end{aligned} \tag{9}$$

where the first equality is the substitution $\underline{z}'=\underline{z}+\underline{e}_Z$, and the last last is due to (3)*. Now, do the bit-flip correction described above for $\mathcal{C}_2^\perp$ (add ancilla, compute syndrome, measure ancilla and discard it, correct $\underline{e}_Z$) to get the state*

$$\sqrt{\frac{|\mathcal{C}_2|}{2^n}}\sum_{\underline{z}'\mathcal{C}_2^\perp}(-1)^{\underline{x}\cdot\underline{z}'^T}\left|\underline{z}'\right\rangle\ , \tag{10}$$

and apply a Hadamard gate on each qubit to get the original state $\left|\underline{x}+\mathcal{C}_2\right\rangle$.

The above proof shows how to *decode* a CSS code, but how do we *encode*? There is a method to encode general quantum error correcting codes, but we wont go into that here (see [1, Section 10.5.5]). Instead, we will present the encoding transformation for the self-dual Hamming-based CSS code. From (4) and the items followed, we deduce that there are M orthogonal states in the CSS code, where M is the number of cosets $\mathcal{C}_2$ has in $\mathcal{C}_1$. If $\mathcal{C}_1=\mathcal{C}_{Ham}$ is the $[n=7,k=4]$ classical Hamming code, and $\mathcal{C}_2=\mathcal{C}_1^\perp$, then it can be shown that there are $M=2$ such cosets. This implies that there are $M=2$ logical states in the code: $|0\rangle_L$ corresponds to the coset $\mathcal{C}_2$ (i.e., $\underline{x}=0$ in (4)),

and $|1\rangle_L$ corresponds to the coset $\underline{1} + \mathcal{C}_2$. By listing the codewords in these cosets we get,

$$\begin{aligned} |0\rangle_L &= \frac{1}{\sqrt{8}} [|0000000\rangle + |0111001\rangle + |1011010\rangle + |1100011\rangle + \\ &\quad |1101100\rangle + |1010101\rangle + |0110110\rangle + |0001111\rangle] \ , \\ |1\rangle_L &= \frac{1}{\sqrt{8}} [|1111111\rangle + |1000110\rangle + |0100101\rangle + |0011100\rangle + \\ &\quad |0010011\rangle + |0101010\rangle + |1001001\rangle + |1110000\rangle] \ . \end{aligned} \tag{11}$$

3 Stabilizer Codes

Stabilizer Codes are a sub-class of quantum error-correcting codes, that are the analogue of the classical linear codes. These codes are based on the stabilizer formalism – a useful tool to understand many quantum operations.

3.1 Stabilizer Formalism

In what follows we denote by A_j an operator that acts non-trivially only on the jth qubit. For example, if $|\psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$, then $X_1 |\psi\rangle = \frac{1}{\sqrt{2}}(|11\rangle + |00\rangle)$.

Definition 4 (Pauli group G_n) *For every $n \geq 1$, the Pauli group of order n is given by*

$$G_n = \{\alpha A_1 \otimes A_2 \otimes \ldots \otimes A_n \ : \alpha \in \{\pm 1, \pm i\}, \ A_j \in \{I, X_j, Y_j, Z_j\}\} \ .$$

Example 7

1. $G_1 = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}$ *(is indeed a group since $Y = iXZ$).*
2. $\frac{1}{\sqrt{2}}(X_1 + Y_1) \notin G_1$.
3. $Z_1 X_3, \ -iI, \ iX_1 Y_2 Z_3 \in G_3$.

Let A be an operator. If for some sate $|\psi\rangle$, $A|\psi\rangle = |\psi\rangle$, then we say that A stabilizes $|\psi\rangle$. For example, $X_1 X_2$ stabilizes the cat state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

Definition 5 (stabilized vector space) *Let S be a sub-group of G_n. The vector space induced by S is defined as*

$$V_S = \{n\text{-qubit states } |\psi\rangle \ : s|\psi\rangle = |\psi\rangle \ , \ \forall s \in S\}.$$

We next use a result from group theory that states that for every finite group G, there exist (at most) $l = \log_2(|G|)$ elements in G, $g_1, g_2, \ldots, g_l$ that *generates* G, in the sense that $G = \{\prod_{k=1}^{l} g_k^{d_k} : d_k \in \mathbb{N}\} \triangleq \langle g_1, g_2, \ldots, g_l \rangle$. The elements, $g_1, g_2, \ldots, g_l$ are called the *generators* of G. Note that to check that a state is stabilized by S, it suffices to check stability by its generators. Furthermore, the dimension of V_S as a vector space, denoted by k is given by $n - l$ (if all generators are independent).

Example 8

1. *Let $n = 3$, and $S = \{I, \ Z_1 Z_2, \ Z_2 Z_3, \ Z_1 Z_3\}$ be a sub-group of G_3. Then, $V_S = \text{span}\{|000\rangle, |111\rangle\}$. Since, $I = (Z_1 Z_2)^2, \ Z_1 Z_3 = (Z_1 Z_2)(Z_2 Z_3)$, then $S = \langle Z_1 Z_2, \ Z_2 Z_3 \rangle$.*

2. *Let $S = \{\pm I, \pm X\}$. Since the only solution to $-I\,|\psi\rangle = |\psi\rangle$ is the zero solution, the vector space V_S is the trivial vector space.*

In view of item 2 above, not every sub-group S is proper (stabilizes a non-trivial vector space). The following Lemma gives necessary and sufficient conditions for a proper sub-group.

Lemma 2 *A sub-group $S = \langle g_1, g_2, \ldots, g_l\rangle \subseteq G_n$ stabilizes a non-trivial vector space if and only if $-I \notin S$ and for every $1 \le i, j \le l$, g_i and g_j commutes, i.e., $g_i g_j = g_j g_i$.*

Example 9 *The $CSS(\mathcal{C}_1, \mathcal{C}_2)$ code with $\mathcal{C}_1$ as the $[7, 4]$ Hamming code from Example 4, and $\mathcal{C}_2 = \mathcal{C}_1^{\perp}$, is a 7-qubit code (vector space) stabilized by the generators*

$$\begin{array}{c|ccccccc} g_1 & X & X & I & X & X & I & I \\ g_2 & X & I & X & X & I & X & I \\ g_3 & I & X & X & X & I & I & X \\ g_4 & Z & Z & I & Z & Z & I & I \\ g_5 & Z & I & Z & Z & I & Z & I \\ g_6 & I & Z & Z & Z & I & I & Z \end{array} \tag{12}$$

It is clear that g_1, g_2 g_3 commute and g_4, g_5 g_6 commute. To see that the other generators commute as well, note that $XZ = -ZX$, and that the overlap between X's and Z's between every to generators is even. Another look at the generators in (12) provides more insights. If we refer the I's as zeros and the X's and Z's as ones, then we can spot the parity-check matrix of $\mathcal{C}_1$ in generators g_1, g_2, g_3 and the parity-check matrix of $\mathcal{C}_2^{\perp}$ (that in this case equals $\mathcal{C}_1$) in generators g_4, g_5, g_6. This is no coincidence, and one can think about the generators as "independent parity constraints" defining the quantum code.

Motivated by Example 9, we now present a useful way of representing the generators. Let $S = \langle g_1, g_2, \ldots, g_l\rangle$ be a sub-group of G_n. Define an $l \times 2n$"*parity-check*" matrix H that represents $g_1, g_2, \ldots, g_l$ as follows. Each row of H corresponds to a generator. For $1 \le i \le l$ and $1 \le j \le n$, let A_{ij} be the effect of generator g_i on qubit j. Then,

$$(H_{i,j}, H_{i,j+n}) = \begin{cases} (0,0) & A_{ij} = I \\ (0,1) & A_{ij} = Z \\ (1,0) & A_{ij} = X \\ (1,1) & A_{ij} = Y \end{cases} \tag{13}$$

It is sometimes convenient to write $H = \left(\begin{array}{c|c} H_x & H_z \end{array}\right)$.

Example 10 *The* "parity-check" *matrix representing the $[n = 7, k = 1]$ stabilizer code from Example 9 is given by*

$$H = \left(\begin{array}{ccccccc|ccccccc} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right) \tag{14}$$

In general, $CSS(\mathcal{C}_1, \mathcal{C}_2)$ codes have a parity matrix with the structure $H = \left(\begin{array}{c|c} H_x & 0 \\ \hline 0 & H_z \end{array}\right)$, and if the underlying codes $\mathcal{C}_1, \mathcal{C}_2$ are duals (as in Example 9) then, $H_x = H_z$.

Lemma 3 (twisted inner product) *Let $H = \left(\begin{array}{c|c} H_x & H_z \end{array} \right)$ be a "parity-check" matrix representing generators $g_1, g_2, \ldots, g_l$. Then, these generators commute if and only if $H_x H_z^T + H_z H_x^T = 0$.*

Example 11 *Parity-check matrices of CSS codes must fulfill*

$$H_x H_z^T = 0 \ , \tag{15}$$

and if they are dual based (as in Example 9), then

$$H_x H_x^T = 0 \ . \tag{16}$$

3.2 Correction Capabilities

Like in their classical counterparts, a quantum code can correct certain errors while there are uncorrectable errors. First, if an error $E \in G_n$ stabilizes the sent state, then it has no effect on it, so in fact it is not an error. More generally, if $S \subset G_n$ defines a stabilizer code and the error $E \in S$, then E is a "no-error" error. Second, if $E \in G_n$ anti-commutes with some $g \in S$, then this error sends the stabilizer code to an orthogonal vector space, and thus is detectable and correctable. The problem is with errors that are not in S but commute with all elements in S.

Theorem 4 *Let S be a sub-group of G_n that defines a non-trivial stabilizer code, and let $Z(S) \triangleq \{E \in G_n \colon Eg = gE, \ \forall g \in S\}$, be the* centralizer *of S. Every $E \notin Z(S) \setminus S$ is correctable.*

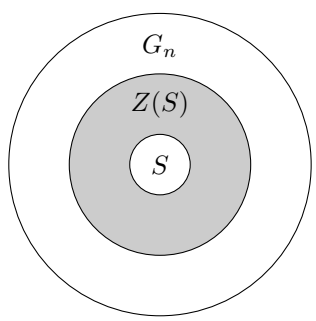

Figure 3: Illustration of the subsets of G_n appearing in Theorem 4. The gray area is the problematic part.

Example 12 *We saw that Shor's code can correct a single arbitrary error by mapping one information qubit into nine encoded qubits. We even improved upon Shor's code with the Hamming-code-based CSS code that can correct a single arbitrary error by mapping one information qubit into seven encoded qubits. Can we improve further? Yes! This 5-qubit code is due to Bennet, Divincenzo, Smoolin and Wooters (96), and is generated by*

$$\begin{array}{c|ccccc} g_1 & X & Z & Z & X & I \\ g_2 & I & X & Z & Z & X \\ g_3 & X & I & X & Z & Z \\ g_4 & Z & X & I & X & Z \end{array} \ . \tag{17}$$

A quick investigation shows that every single-qubit error $E \in G_5$ anti-commutes with all generators (e.g., $X_1 g_4 = -g_4 X_1$, $Z_2 g_2 = -g_2 Z_2$), which in view of Theorem 4 implies that the code can correct an arbitrary single error. Can we improve upon this 5-qubit code? No! It can be shown via the quantum Hamming bound that 5 qubits are the minimum needed.

4 Q-LDPC Codes

We now move to a specific family of stabilizer code: Quantum-LDPC codes. These codes are an extension of their celebrated classical predecessors – LDPC codes. Hence, in order to understand Q-LDPC codes, a tour over LDPC codes should be made; we move back to the classical regime.

4.1 LDPC Codes

A linear code is an LDPC (low-density parity-check) code of it has a parity-check matrix H that is sparse. This means that the number of non-zero elements H is linear in the code length n, i.e., $O(n)$. The most attractive property of LDPC codes, is that they show excellent performance with low redundancy with very-low-complexity decoding algorithm. Decoding LDPC code is done in an iterative fashion, and is based on a graphical representation of these codes. For example, consider the Hamming code of order $m = 3$ from Example 4. Writing each row in the equation $H\underline{c}^T = 0$ yields

$$\begin{array}{ccccccccc} c_1+ & c_2+ & & c_4+ & c_5 & & & = & 0 \\ c_1+ & & c_3+ & c_4+ & & c_6 & & = & 0 \\ & c_2+ & c_3+ & c_4+ & & & c_7 & = & 0. \end{array} \tag{18}$$

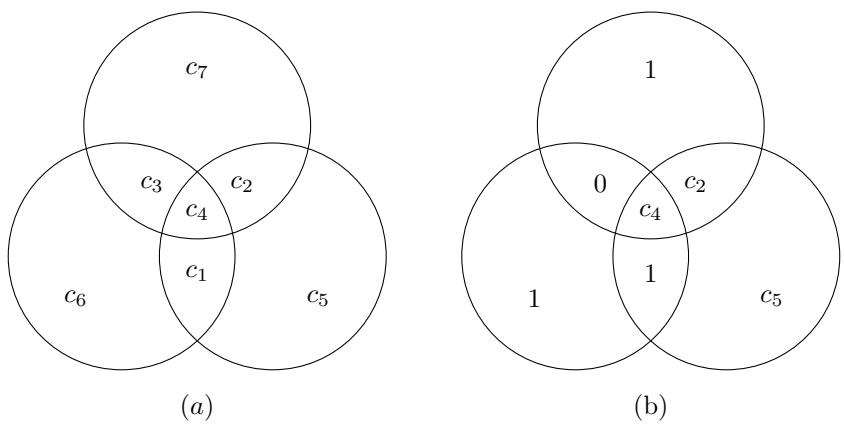

Figure 4: (a) a Venn diagram representing the parity constraints in (18); (b) the decoding problem.

These parity constraints are graphically presented in a *Venn* diagram in Figure 4(a). Each constraint is a circle containing the bits that are constrained by it. On this diagram, the iterative decoding algorithm can be easily explained. Suppose we know that $c_1 = 1,\ c_3 = 0,\ c_6 = 1,\ c_7 = 1$, and we want to reconstruct $\underline{c}$. Then, we have the diagram in Figure 4(b). From the left-most constraint we can see that $c_4 = 0$. After solving c_4, we can now use the upper constraint to get $c_2 = 1$. Finally, with the left-most constraint we get $c_5 = 0$. This decoding procedure is known as iterative decoding, and it is part of many technological standards nowadays. The problem with Venn diagrams is that they become messy with more constraints. We need a different representation. Instead of Venn diagrams, we will use *Tanner* graphs.

Definition 6 *Let H be a parity-check matrix of some linear code. The Tanner graph representing H is an un-directed bipartite graph $(V \cup C,\ E)$, such that V is the set of bit nodes (variable nodes) indexed by the columns of H, and C is the constraint nodes (check nodes) indexed by the rows of H. For every $v \in V,\ c \in C$, there exist an edge $e = (v, c)$ if the corresponding entry in H is non-zero.*

Example 13 *The Tanner graph representing the parity-check matrix in* (1) *is given in Figure 5. The bit nodes V are circles. and the constraint nodes C are squares.*

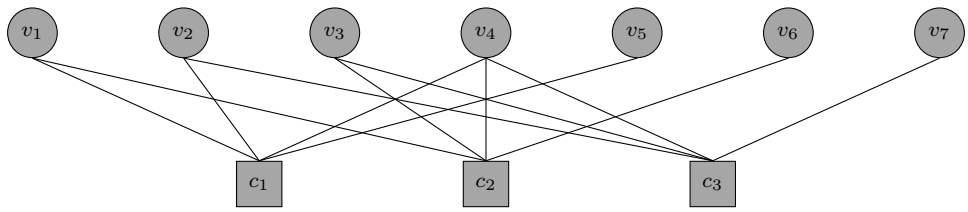

Figure 5: Tanner graph representing the parity-check matrix in (1).

The decoding algorithms of LDPC codes are performed over their Tanner graphs. In these algorithms, messages are passed on edges between the graph's nodes in an iterative fashion, a.k.a message-passing algorithms. At the end of the process, the decoder outputs a codeword that was likely to be sent (an estimation). There are many types of message-passing algorithms, and the optimal one is called Belief Propagation (BP). In these algorithms every message passing on an edge is a belief concerning the value of the bit connected to this edge. This means that the alphabet used by this algorithm is not binary (mathematically, the messages are $\mathbb{R}$). The decoding abilities of LDPC codes strongly depend on the Tanner graph's structure. In particular, short cycles are known to have a very bad influence.

Example 14 (LDPC Codes in QKD) *As efficient and powerful error-correcting codes, LDPC codes can be used in the* information reconciliation *step of continuous-variables quantum key distribution. We won't go into this application here (see [3]).*

4.2 Q-LDPC Codes

We can draw a Tanner graph for any stabilizer code similarly to the classical case. For example, the parity matrix of CSS codes has a block structure $H = \left(\begin{array}{c|c} H_x & 0 \\ \hline 0 & H_z \end{array}\right)$, so we can draw two independent Tanner graphs: one for the X errors and the other for the Z errors. For general stabilizer codes we can draw a *labeled* Tanner graph, where each constraint node c_i corresponds to a generator g_i, and an edge between c_i and a variable node v_t exist if g_i affects non-trivially on qubit t. An edge $e = (v_t, c_i)$ is labeled by the corresponding Pauli operator X, Z or Y. Figure 6 illustrates the Tanner graph of the 5-qubit code in (17).

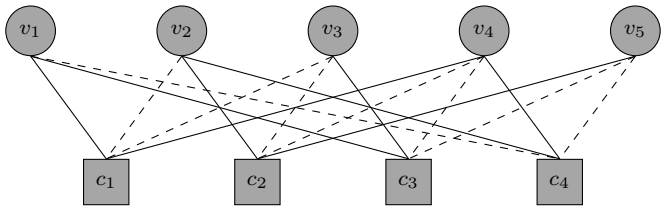

Figure 6: The Tanner graph of the 5-qubit code in (17). Solid (resp. dashed) edges correspond to an X (resp Z) in the generator.

Clearly, every qubit should be protected against bit-flip and phase-flip errors. Thus, each qubit should be involved at least in two generators with different non-identity Pauli operators. In addition, due to the commutativity constraint in Lemma 2, every pair of rows must have an even number of qubit indices with different non-identity Pauli operators. Consequently, all QLDPC Tanner graphs have cycles of length 4. From the theory on classical LDPC codes, we know that such short cycles have very a bad impact on decoding. There are several solutions to this problem, such as: 1) *entangelment-assited* codes where some prepared entangled qubits are shared between the transmitter and receiver (like dense-coding); 2) error injection into the decoding process. We won't go into these variations here.

Decoding Q-LDPC codes is similar to the classical case, with the difference that one cannot measure the qubits. Instead, the syndrome is measured (using ancilla) and then the syndrome is fed into a syndrome-based classical LDPC decoder that estimates the error from the channel (i.e. estimated $E \in G_n$).

References

[1] M. A. Nielsen and I. L. Chuang "Quantum computation and quantum information", *Cambridge, university press*, 2010.

[2] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng and L. Hanzo, "Fifteen Years of Quantum LDPC Coding and Improved Decoding Strategies," *IEEE Access*, vol. 3, pp 2492-2519, Nov. 2015.

[3] J. Lodewyck, M. Bloch, R. García-Patrón, S. Fossier, EKarpov, E. Diamanti, T. Debuisschert, Thierry, N. J. Cerf, R. Tualle-Brouri, S. W. McLaughlin, and P. Grangier, "Quantum key distribution over 25 km with an all-fiber continuous-variable system," *Phys. Rev. A*. vol. 76, no. 4, Oct. 2007.