

# Quantum Information Theory - 67749

## Recitation 3, June 4, 2025

## 1 Recitation Overview - Quantum Error Correction Codes.

In the last lectures, we introduced quantum channels - physical maps between density matrices and mentioned that they are used to represent noise, an error that might occur during either computation or communication. Faults and errors are not unique to the quantum regime and were studied before in the context of (classical) error correction codes and fault tolerance. In this recitation, we are first going to represent what classical error correction codes are, and we will give the Tanner codes as an example of the construction of such codes. Then we will talk about the standard way to model quantum noise, give a basic example of a first quantum code, and then eventually we will give the Toric code.

### Part I

## Classical Error Correction Code.

## 2 Codes in General. Notations and Definitions.

A code is a subset, which one can think of as the valid points. Here we focus only on linear binary codes, which are linear subspaces of  $\mathbb{F}_2^n$ . Requiring linearity allows us to talk about efficiency. Now, our subspace is the image of the encoding matrix or the kernel of the checking matrix. As implied at the beginning, codes are used to correct errors, a correctable error is one that gets a valid point into its local environment. A common way to measure resilience is to ask how many bits an evil entity needs to flip such that the corrupted vector will be closer to another vector in that space than the original one. Those ideas were formulated by Hamming [Ham50], who presented the following definitions.

**Definition 2.1.** Let  $n \in \mathbb{N}$  and  $\rho, \delta \in (0, 1)$ . We say that  $C$  is a **binary linear code** with parameters  $[n, \rho n, \delta n]$ . If  $C$  is a subspace of  $\mathbb{F}_2^n$ , and the dimension of  $C$  is at least  $\rho n$ . In addition, we call the vectors belong to  $C$  codewords and define the distance of  $C$  to be the minimal number of different bits between any codewords pair of  $C$ .

Moreover, since  $C$  is a linear subspace, there are matrices  $G : \mathbb{F}_2^{\rho n} \rightarrow \mathbb{F}_2^n$  and  $H : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^s$  such that  $C = \text{image } G$  and  $C = \ker H$ .  $G$  is called the generator matrix of  $C$ , while  $H$  is called the parity check matrix of  $C$ .

From now on, we will use the term code to refer to linear binary codes, as we don't deal with any other types of codes. Also, even though it is customary to use the above parameters to analyze codes, we will use their percent forms called the relative distance and the rate of code, matching  $\delta$  and  $\rho$  correspondingly.

**Definition 2.2.** A **family of codes** is an infinite series of codes. Additionally, suppose the rates and relative distances converge into constant values  $\rho, \delta$ . In that case, we abuse the notation and call that family of codes a code with  $[n, \rho n, \delta n]$  for fixed  $\rho, \delta \in [0, 1)$ , and infinite integers  $n \in \mathbb{N}$ .

Notice that the above definition contains codes with parameters attending to zero. From a practical view, it means that either we send too many bits, more than a constant amount, on each bit in the original message. Or that for big enough  $n$ , adversarial, limited to changing only a constant fraction of the bits, could disrupt the transmission. That distinction raises the definition of good codes.

**Definition 2.3.** We will say that a family of codes is a **good code** if its parameters converge into positive values.

**Example 2.1** (The Check-sum/Even-parity Code.). The parity check code  $C_p \subset \mathbb{F}_2^{n+1}$  consists of codewords that are vectors with even parity. Alternatively, it can be described as appending the xor of the original codeword at the end, sometimes called a checksum bit.

$$\begin{aligned} 101 &\mapsto 1010 \\ 111 &\mapsto 1111 \\ 000 &\mapsto 0000 \end{aligned}$$

**Example 2.2** (The Repetition Code.). Define the repetition code  $C_r \subset \mathbb{F}_2^{n \cdot r}$ , In which, for a fixed integer  $r$ , any bit of the original string is duplicated  $r$  times. For example:

$$\begin{aligned} 0 &\mapsto 0000 \\ 1 &\mapsto 1111 \end{aligned}$$

Having both examples in mind, we get a feeling of the behavior of the distance-rate trade-off. To summarize, using a simple construction, one could construct the codes  $[r, 1, r]$  and  $[r, r - 1, 2]$ . Each has a single perfect parameter, while the other decays to the worst. That's bring us to present the Singleton Bound.

Besides being the first bound, Singleton bound demonstrates how one could get results by using relatively simple elementary arguments. It is also engaging to ask why the proof yields a bound that, empirically, seems far from being tight.

**Theorem** (Singleton Bound.). *For any linear code with parameter  $[n, k, d]$ , the following inequality holds:*

$$k + d \leq n + 1$$

*Proof.* Since any two codewords of  $C$  differ by at least  $d$  coordinates, we know that by ignoring the first  $d - 1$  coordinate of any vector, we obtain a new code with one-to-one corresponding to the original code. In other words, we have found a new code with the same dimension embedded in  $\mathbb{F}_2^{n-d+1}$ . Combine the fact that dimension is, at most, the dimension of the container space, we get that:

$$\dim C = 2^k \leq 2^{n-d+1} \Rightarrow k + d \leq n + 1$$

□

It is also well known that the only binary codes that reach the bound are:  $[n, 1, n]$ ,  $[n, n - 1, 2]$ ,  $[n, n, 1]$  [AF22]. In particular, there are no good binary codes that obtain equality (And no binary code which get close to the equality exists).

Next, we will review Tanner's construction, that in addition to being a critical element to our proof, also serves as an example of how one can construct a code with arbitrary length and positive rate.

Another parameter that will interest us is the complexity of performing checks. Consider a computation model in which the bits are exposed to errors at any time step. In those models, a long check might accumulate faults while being performed. So, if one would like to assume that with non-vanishing probability the check gives a true result, then he has to ensure that it takes constant time.

**Definition 2.4** (LDPC codes.). *We will say that a code  $C$  is a **low density parity check matrix** code if any check of it (a row in the parity check matrix), checks touch at most  $O(1)$  bits.*

**Exercise 2.1.** *Give two representations of the checking matrix of the repetition code, one which is LDPC and the other which is not.*

### 3 Tanner Code

The constructions require two main ingredients: a graph  $\Gamma$ , and for simplicity, we will restrict ourselves to a  $\Delta$  regular graph. Yet notice that the following could be generalize straightforwardly for graphs with degree at most  $\Delta$ . The second ingredients is a 'small' code  $C_0$  at length equals the graph's regularity, namely  $C_0 = [\Delta, \rho\Delta, \delta\Delta]$ . We can think about any bit string at length  $\Delta$  as an assignment over the edges of the graph. Furthermore, for every vertex  $v \in \Gamma$ , we will call the bit string, which is set on its edges, the local view of  $v$ . Then we can define, [Tan81]:

**Definition 3.1.** *Let  $C = \mathcal{T}(\Gamma, C_0)$  be all the codewords which, for any vertex  $v \in \Gamma$ , the local view of  $v$  is a codeword of  $C_0$ . We say that  $C$  is a **Tanner code** of  $\Gamma, C_0$ . Notice that if  $C_0$  is a binary linear code, So  $C$  is.*

**Example 3.1.** Consider the Petersen graph  $\Gamma$ , which is a regular graph with degree 3. Let  $C_0$  be the set of all words with even parity. It follows that  $C_0$  contains all even-length binary strings of length 3: 000, 110, 101, and 011. However, the size of  $\mathcal{T}(\Gamma, C_0)$  is significantly larger, as shown in Figure fig. 1. Specifically, any rotation of the inner and outer cycles simultaneously gives rise to another valid codeword, so any assignments that are not invariant under these rotations would produce five additional valid codewords.

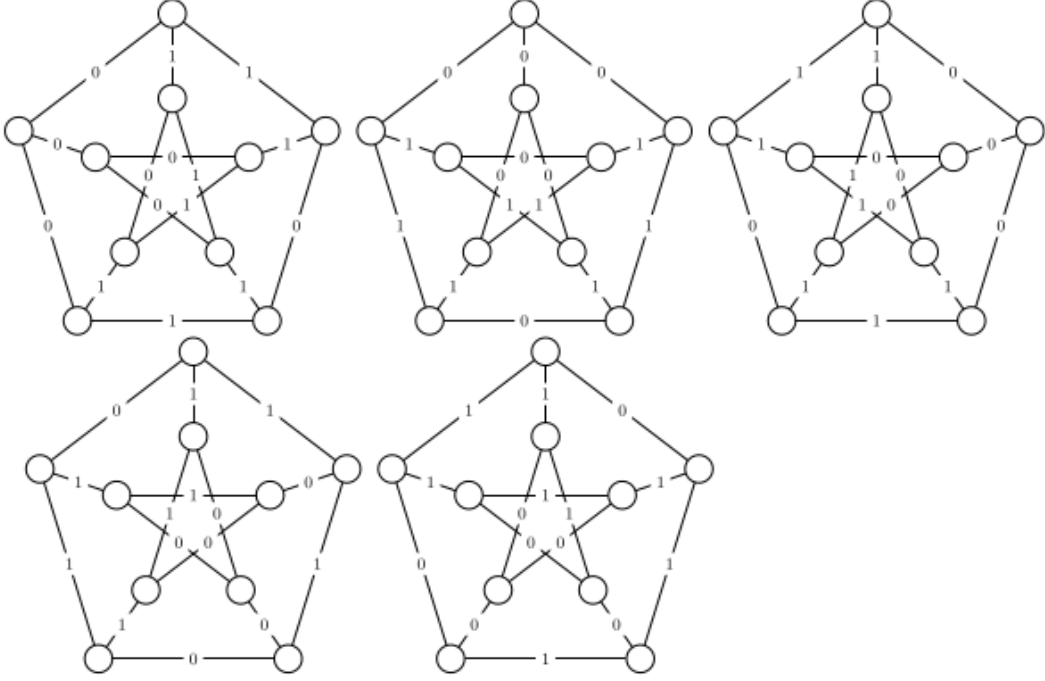


Figure 1: Peterson Graph.

**Lemma 3.1.** Tanner codes have a rate of at least  $2\rho - 1$ .

*Proof.* The dimension of the subspace is bounded by the dimension of the container minus the number of restrictions. So assuming non-degeneration of the small code restrictions, we have that any vertex count exactly  $(1 - \rho) \Delta$  restrictions. Hence,

$$\dim C \geq \frac{1}{2}n\Delta - (1 - \rho) \Delta n = \frac{1}{2}n\Delta (2\rho - 1)$$

Clearly, any small code with rate  $> \frac{1}{2}$  will yield a code with an asymptotically positive rate  $\square$

Based on Lemma 3.1, we can obtain a recipe for constructing codes with a almost non-vanishing rate for arbitrarily large lengths and dimensions. This recipe involves

concatenating a series of Tanner codes over complete graphs. To be more precise, we can define a family of codes as follows:

$$\begin{aligned} C_{i+1} &= \mathcal{T}(K_{n(C_i)+1}, C_i) \\ C_0 &= \text{Some simple } \Delta[1, \rho_0, \delta_0] \text{ code.} \end{aligned}$$

Where  $n(C_i)$  represents the code length of the  $i$ th code. Repeating the process described above  $\log_{\Delta}^*(n)$  times allows us to extend the initial code  $\Delta[1, \rho_0]$  to  $n[1, \sim 2\rho^{\log_{\Delta}^*(n)}]$ . Interestingly, any family of finite groups generated by a constant-size generator set can define a family of codes by utilizing their Cayley graphs as a basis for Tanner codes.

Once we have seen that Tanner codes enable us to achieve rates, the next natural question to ask is about the distance of the codes. Achieving a linear distance requires a little bit more from the graphs, but to understand this idea better, let us return to the repetition code. For instance, the repetition code can be presented as a Tanner code over the cycle graph.

**Example 3.2.** *In this representation, each vertex checks if the bits on its edges are equal. A valid codeword is an assignment in which all the bits are equal, since otherwise, there would be an edge with no supporting vertex. An illustration of a legal assignment is provided in section 3.*

Recall that the distance of a linear code is the minimal weight of the non-zero codewords. Consider a codeword  $c \in C$  and group the vertices by four sets  $V_i$  such that  $V_i$  is the set of vertices that see  $i \in \{0, 1\}^2$ . Since  $c \in C$ , we have that  $|V_{10}| = |V_{01}| = 0$ . Additionally, any vertex in  $V_{00}$  is not connected to  $V_{11}$ , which gives us two possible cases: either all the vertices in  $V_{11}$  are isolated, or the graph is not connected. Hence, the distance of the code is equal to  $\frac{1}{2} \sum |V_i| \cdot |i| = \frac{1}{2} 2 \cdot n = n$ .

It is worth mentioning that, in the literature, the repetition code is not usually given as an example of a Tanner code. However, this example will come up again later in the chapter on quantum codes, when we discuss the Toric code, its relation to the hyperproduct code, and how it can be seen as a hyperproduct of two cycle codes.

Furthermore, analyzing the repetition code gives a clue as to how, in certain cases, one might prove a lower bound on the code distance. We would like to say that, if the weight of the code word is below the distance, then it must be that there is at least one vertex that has a non-trivial local view which is not a codeword in  $C_0$ . Put differently, we cannot spread a small weight codeword over  $\{V_i\}$ , defined above, without expanding into subsets corresponding to low  $|i|$ . Next, we are going to present the Expander codes, which are Tanner codes constructed from graphs with good algebraic expansion.

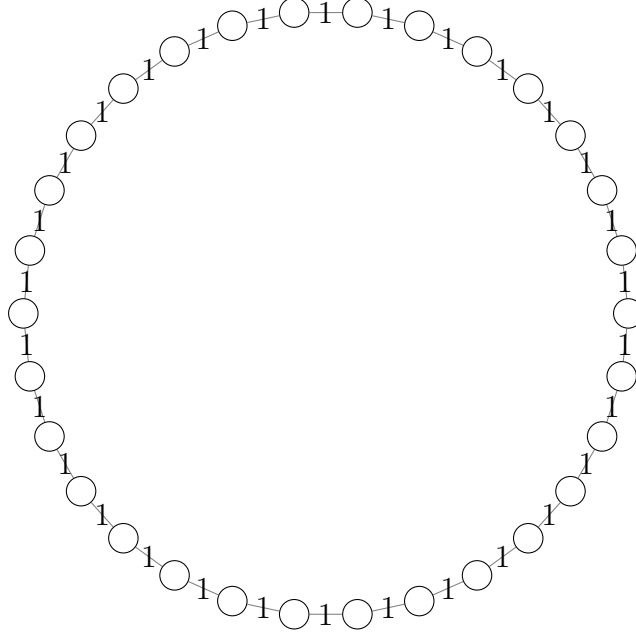


Figure 2: The  $1^n$  assignment on the cycle graph. Any vertex compute parity  $1+1 = 0$ , therefore all the restrictions are satisfied and  $1^n \in \mathcal{T}(\text{cycle}, \text{parity})$ .

## Part II

# Quantum Error Correction Codes.

## 4 Introduction.

The second difference, which is a subtle point, is that quantum states are susceptible to additional types of errors. In addition to the possibility of bit-flip errors, a quantum state may also experience a change in phase. For example, consider the initial state  $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ , and suppose that due to noise the state is transformed into  $\frac{1}{\sqrt{4}}(\sqrt{3}|0\rangle + |1\rangle)$ . Classical circuits are oblivious to such faults, meaning that their operation would remain unchanged as no error has occurred. Quantum circuits, however, are usually affected and may fail. Furthermore, when designing a decoder for quantum error correction codes, one must ensure that the decoding process does not introduce bit-flip errors if a classical code is used to protect against phase flips.

## 5 Quantum Noise.

**Definition 5.1** (Bit and phase flip.). *Consider a quantum state  $|\psi\rangle$  encoded in the computation base. We will say that a bit-flip occurs in a scenario the operator Pauli  $X$  is applied on one of our state's qubits. The bit-flip event could be considered as exactly as the standard bit-flip error in the classical regime. Similarly, phase-flip*

occurs when the Pauli  $Z$  is applied on one of the qubits.

Let's return to the repetition code presented in Chapter 2. We would like to have an analog; a first and natural attempt might consider duplicating copies of the state. Unfortunately, copying a general state is not a linear operation and therefore cannot be done in the circuit model (or any other believed to be feasible). In particular, there is no circuit  $U$  which can simultaneously duplicate the states  $|0\rangle, |1\rangle, |+\rangle, |-\rangle$ .

To overcome the issue, Shor came up with the nine-qubit code [Sho95], which at first glance might seem a naive straightforward implementation of “duplication”, but instead uses a clever insight about quantumness in general. Any operation can be seen as a linear (and even unitary) operation over a subspace embedded in a large enough dimension. The encoding is given as follows:

$$\begin{aligned} |\bar{0}\rangle &= \frac{1}{2\sqrt{2}} (|000\rangle + |111\rangle)^{\otimes 3} \\ |\bar{1}\rangle &= \frac{1}{2\sqrt{2}} (|000\rangle - |111\rangle)^{\otimes 3}. \end{aligned}$$

For convenience, let us use the notation  $|\mathbf{GHZ}^\pm\rangle = |0^m\rangle \pm |1^m\rangle$ . We can also consider the Shor code over  $m^2$  qubits, which is defined as above, such that any logical state contains  $m$  products over  $m$  qubits. Therefore, the state  $|\bar{0}\rangle$  over  $m^2$  qubits can be written as  $|\mathbf{GHZ}^+\rangle^m$ . We are now ready to prove a statement regarding the robustness.

**Lemma 5.1.** *The Shor code over 9 qubits enable to correct a single either bit or phase flip.*

It is evident that a single bit-flip error can be handled in the same way as in the conventional case. The decoder will check if any of the triples have the same value, and if not, it will correct it by majority. To create a decoder that can also correct a phase-flip error, we need the following statement. In this chapter, we denote the Hadamard gate over  $m$  qubits as  $H^m$ .

**Claim 5.1.**  $H^m |\mathbf{GHZ}^\pm\rangle = \sum_{x \cdot \mathbf{1} = 2^\pm} |x\rangle$

*Proof.*

$$\begin{aligned} H^m |\mathbf{GHZ}^\pm\rangle &= H^m |0^m\rangle \pm H^m |1^m\rangle = \sum_{x \in \mathbb{F}_2^m} |x\rangle \pm \sum_{x \in \mathbb{F}_2^m} (-1)^{x \cdot \mathbf{1}} |x\rangle \\ &= \sum_{x \in \mathbb{F}_2^m} (1 \pm (-1)^{x \cdot \mathbf{1}}) |x\rangle = \sum_{x \cdot \mathbf{1} = 2^\pm} |x\rangle \end{aligned}$$

□

Now it is clear how to correct a phase flip. One can apply the Hadamard transform and compute the parity of each triple. By the assumption that only a single phase flip may occur, either all the triples have the same parity or the faulted one has an opposite parity and needs to be corrected. Thus, we obtain an  $[[9, 1, 3]]$  quantum error correction code. Asymptotically, this is an  $[[m^2, 1, m]]$  code.

## 6 CSS Codes.

The Shor code is a specific case of the more general CSS (Calderbank-Shor-Steane) code [CS96]. A family composed by two binary codes  $C_X, C_Z$  such that  $C_Z^\perp \subset C_X$ .

**Definition 6.1** (CSS Code). *Let  $C_X, C_Z$  classical linear codes such that  $C_Z^\perp \subset C_X$  define the  $Q(C_X, C_Z)$  to be all the code words with following structure:*

$$|\mathbf{x}\rangle := |x + C_Z^\perp\rangle = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} |x + z\rangle$$

Clearly, the codewords are all the codewords in  $C_X$  which don't belong to  $C_Z^\perp$  and therefore the dimension of the quantum code is  $\dim Q(C_X, C_Z) = \dim C_X - \dim C_Z^\perp = \dim C_X + \dim C_Z - n$ . Yet, it's not stems immediately how one can correct faults. Next, we are going to repeat the decoding process of the Shor code in the general setting of CSS codes.

**Lemma 6.1.** *Let  $C_X, C_Z$  classical codes such  $Q(C_X, C_Z)$  is a CSS code. Let  $d_X$  be the minimal weigh of codeword in  $C_X$  which is not in  $C_Z^\perp$ , and define by the same way  $d_Z$  to be the minimal weight of codeword in  $C_Z$  which doesn't belong to  $C_X^\perp$ . Then the distance of  $Q(C_X, C_Z)$  equals to  $\min d_X, d_Z$ . Moreover there is a decoder which correct any fault with weight at most  $d/2$ .*

*Proof.* First let us prove the following claim:

**Claim 6.1.** *Denote by  $H^{\otimes n}$  the Hadamard gate over  $n$  qubits. Then for any code  $C$  it holds that:  $H^n |C^\perp\rangle = |C\rangle$*

*Proof.*

$$\begin{aligned} H^n |C^\perp\rangle &= \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} H^n |z\rangle = \frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} \sum_{y \in \mathbb{F}_2^n} (-1)^{\langle z, y \rangle} |y\rangle \\ &= \frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} \left( \sum_{y \in C_Z} |y\rangle + \text{other terms} \right) \end{aligned} \tag{1}$$

Since the columns of matrix  $H_Z$  form a basis for the complementary space  $C_Z^\perp$ , and due to the dimensional theorem and the equivalence between the row rank and column rank of a matrix, we can deduce that  $\dim \text{rank } H_Z^\top + \dim \ker H_Z = n$ , which implies that  $|C_Z^\perp| |C_Z| = 2^n$ . Thus the norm of

$$\frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} \sum_{y \in C_Z} |y\rangle = \sqrt{\frac{|C_Z^\perp|}{2^n}} \sum_{y \in C_Z} |y\rangle \tag{2}$$

equals 1 and the summation over the vectors  $y \notin C_Z$  in the inner closure in equation 1 must cancel. So we left only with a uniform superposition over the codewords in  $C_Z$ , Or in other words  $H^n |C^\perp\rangle = |C\rangle$ .  $\square$



Claim 6.1 states that, when considering CSS codes, pauli  $X$  operators can be seen as pauli  $Z$  operators in the rotated frame. That it,

$$H^n X^f |C_Z^\perp\rangle = \overbrace{H^n X^f H^n}^{Z^f} H^n |C_Z^\perp\rangle = Z^f |C\rangle$$

That insight hints a description to decoder for the quantum code. If one knows how to correct errors for each of the classical code  $C_X, C_Z$  than he can start by correct the bit flips in using the decoder of  $C_X$ , rotate the state by applying the Hadamard transform and then correct, what was before the transformation phase flips and now a bit flips by using the decoder of  $C_Z$ . Then in the end applying the Hadamard transform again for backing to the initial computation space. Indeed that decoder correct  $\min\{d(C_X)/2, d(C_Z)/2\}$  errors. Yet more work is needed to show that this decoder also correct  $d(Q(C_X, C_Z))/2$  errors.

Let us assume the existences of decoders of the classical codes  $C_X$  and  $C_Z$ , Denoted  $D_X : \mathbb{F}_2^n \rightarrow C_X$  and  $D_Z$ . In particular for any  $\xi \in \mathbb{F}_2^n$ ,  $D_X(\xi) = \arg \min_{x \in C_X} |x + \xi|$ .

**Claim 6.2.** *For any  $x_0 \in C_X$  and  $z_1 \neq z_2 \in C_Z^\perp$ ,  $D_X$  correct  $|x + z_1 + f\rangle, |x + z_2 + f\rangle$  into two different words in  $C_X$ .*

*Proof.* Suppose not, namely there exists  $y \in C_X$  such that  $D_X$  correct  $|x + z_1 + f\rangle, |x + z_2 + f\rangle$  into  $|y\rangle$ . Then we have that for both  $i \in \{1, 2\}$  it holds that  $d(x + z_i + f, y) \leq d(C_Z^\perp/2)$  and therefore  $d(x + z_1 + f, x + z_2 + f) \leq d(C_Z^\perp)$ . But

$$\begin{aligned} d(x + z_1 + f, x + z_2 + f) &= |x + z_1 + f + x + z_2 + f| \\ &= |z_1 + z_2| = d(z_1, z_2) \end{aligned}$$

contradiction for the assumption that  $z_1, z_2 \in C_Z^\perp$ .  $\square$

We are ready to show step by step the decoding process. Let  $P = X^f Z^e$  be an error such that  $e, f < d/2$  act on the state  $|\mathbf{x}\rangle$ . Denote by  $H_X, H_Z$  the parity check matrices of  $C_X, C_Z$ . Using the commute relation  $[X^f, Z^e] = (-1)^{\langle e, f \rangle}$  we have that:

$$|\mathbf{x}\rangle \mapsto^P \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^f Z^e |x + z\rangle = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} (-1)^{\langle e, f \rangle} Z^e |x + z + f\rangle$$

Now the decoder computes and stores the syndrome relative to the bits code using the parity check matrix  $H_X$ . And apply the inverse gate.

$$\begin{aligned} &\mapsto^{H_X} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} (-1)^{\langle e, f \rangle} Z^e |x + z + f\rangle |H_X(x + z + f)\rangle \\ &= \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} (-1)^{\langle e, f \rangle} Z^e |x + z + f\rangle |H_X f\rangle \\ &\mapsto^{X^f} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} Z^e |x + z\rangle |H_X f\rangle \end{aligned}$$

Then rotating into the phases base:

$$\begin{aligned}
& \mapsto^{H^{\otimes n}} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^e H^{\otimes n} |x + z\rangle \\
&= \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^e H^{\otimes n} X^x |z\rangle = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^e Z^x H^{\otimes n} |z\rangle \\
&= \frac{1}{\sqrt{|C_Z^\perp|}} X^e Z^x H^{\otimes n} \sum_{z \in C_Z^\perp} |z\rangle = \frac{1}{\sqrt{|C_Z|}} X^e Z^x \sum_{z \in C_Z} |z\rangle \\
&= \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x, e \rangle} Z^x \sum_{z \in C_Z} X^e |z\rangle = \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x, e \rangle} Z^x \sum_{z \in C_Z} |z + e\rangle
\end{aligned}$$

So now we have back into the begging. Only now the phase flips are playing the role of bit flips relative to the code  $C_Z$ .

$$\begin{aligned}
&= \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x, e \rangle} Z^x \sum_{z \in C_Z} |z + e\rangle |H_Z e\rangle \\
&\mapsto \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x, e \rangle} Z^x \sum_{z \in C_Z} |z\rangle |H_Z e\rangle \mapsto^{H^n} \frac{1}{\sqrt{|C_Z^\perp|}} X^x \sum_{z \in C_Z^\perp} |z\rangle \\
&= \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} |z + x\rangle = |\mathbf{x}\rangle
\end{aligned}$$

□

There is still one big difference between the classic repetition code and the Shor code. While each parity check of the Shor code examines a square root number of qubits, any check of the repetition code touches no more than a constant number of qubits; that is, any check just tests if any two adjacent bits are equal. That brings us to ask whether the Shors code is really the quantum analogy for the repetition code?

For getting an hint before formally presenting a quantum LDPC code, let's take another look on the general structure of the CSS codes. The decoding procedure the proof above teach us an additional point about CSS code, the task of finding a good code quantum code, could be reduce for finding a two classic binary linear codes which their parity check matrices ortogonal to each other. Furthermore, if one is willing to has an qLDPC code, then  $H_X$  and  $H_Z$  can't be parity check matrices of good classical code as any column of  $H_Z^\top$  is a codeword of  $C_X$ .

$$C_Z^\perp \subset C_X \Rightarrow H_X H_Z^\top = 0$$

And by being an LDPC code, the rows wights of  $H_Z$  is bounded by constant. Therefore there is a codeword  $\in C_X$  which is also a row of  $H_Z$  that has a constant weight.

## 7 qLDPC Codes.

As exactly as in the classic case, qLDPC codes are codes in which any check act non trivially on at most a constant number of qubits, It was proved that using a good Quantum LDPC code one can achieve a fault tolerance threshold theorem at the cost of only constant overhead<sup>1</sup> [Got14] in space. We are now about to embark on a review of the first quantum LDPC code [Den+02].

Recall that one way to present a code is by define the parity check matrix, Consider the  $l \times l$  Tours, namely the Cayley graph of the group product  $\mathbb{Z}_l \times \mathbb{Z}_l$ . Associate any coordinate (bit/qubit) with an edge on the Tours. And consider the following two restrictions:

1. Each vertex requires form its local view, the bits lay on its supported edges, To has an even parity. We will refer to this type of check as *cross check*. Namely  $C_X$  is the tanner code defined by use the Toric as a graph and the parity code as the local code.
2. Similarly, each face requires the same from its supported edges but computes the parity in a phase basis, namely the  $C_Z$  code defined again as Tanner code, when the graph is defined by associating with each face a vertex and connecting any two adjacent faces by an edge, which corresponds to the side of the square.

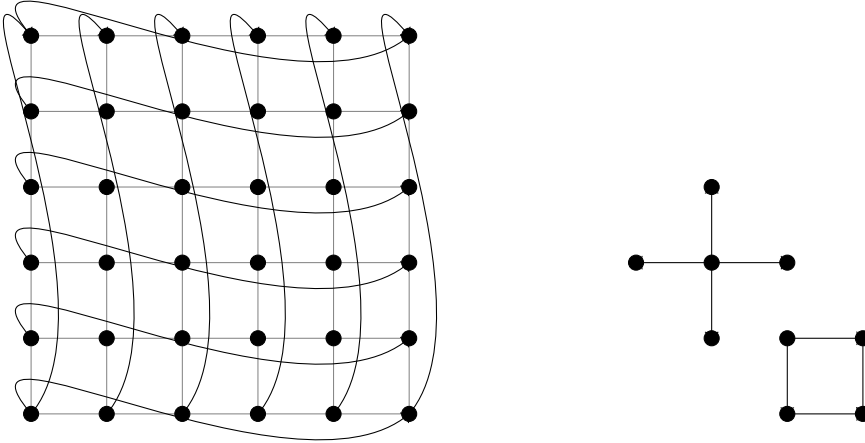


Figure 3: On the left is the Toric Graph. On the right are cross and face checks.

**Claim 7.1.** *The  $l \times l$  Toric code is a CSS code, with dimension 2 and distance  $\Theta(l)$ .*

---

<sup>1</sup>under the assumption of having an efficient decoder.

## References

- [Ham50] R. W. Hamming. “Error detecting and error correcting codes”. In: *The Bell System Technical Journal* 29.2 (1950), pp. 147–160. DOI: [10.1002/j.1538-7305.1950.tb00463.x](https://doi.org/10.1002/j.1538-7305.1950.tb00463.x).
- [Tan81] R. Tanner. “A recursive approach to low complexity codes”. In: *IEEE Transactions on Information Theory* 27.5 (1981), pp. 533–547. DOI: [10.1109/TIT.1981.1056404](https://doi.org/10.1109/TIT.1981.1056404).
- [Sho95] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Phys. Rev. A* 52 (4 Oct. 1995), R2493–R2496. DOI: [10.1103/PhysRevA.52.R2493](https://doi.org/10.1103/PhysRevA.52.R2493). URL: <https://link.aps.org/doi/10.1103/PhysRevA.52.R2493>.
- [CS96] A. R. Calderbank and Peter W. Shor. “Good quantum error-correcting codes exist”. In: *Physical Review A* 54.2 (Aug. 1996), pp. 1098–1105. DOI: [10.1103/physreva.54.1098](https://doi.org/10.1103/physreva.54.1098). URL: [https://doi.org/10.1103%2Fphysreva.54.1098](https://doi.org/10.1103/2Fphysreva.54.1098).
- [Den+02] Eric Dennis et al. “Topological quantum memory”. In: *Journal of Mathematical Physics* 43.9 (Sept. 2002), pp. 4452–4505. DOI: [10.1063/1.1499754](https://doi.org/10.1063/1.1499754). URL: <https://doi.org/10.1063%2F1.1499754>.
- [Got14] Daniel Gottesman. *Fault-Tolerant Quantum Computation with Constant Overhead*. 2014. arXiv: [1310.2984](https://arxiv.org/abs/1310.2984) [quant-ph].
- [AF22] “Maximum distance separable (MDS) code”. In: *The Error Correction Zoo*. Ed. by Victor V. Albert and Philippe Faist. 2022. URL: <https://errorcorrectionzoo.org/c/mds>.