1 Setup

In today's lecture, we will continue the discussions on quantum error-correcting codes. The reading materials for this lecture are the introductory exposition by Roffe [Rof19] and John Preskill's notes [Pre].

Let us quickly recall the setup. We have defined the Pauli operators

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{24.1}$$

Define the basis vectors of a single-qubit space $\mathbb{C}^2$ as $|0\rangle = \begin{bmatrix} 1 & 0 \end{bmatrix}^\top, |1\rangle = \begin{bmatrix} 0 & 1 \end{bmatrix}^\top$. Then, X represents a bit-flip error. Namely, $X|b\rangle = |1-b\rangle$ for $b \in \{0,1\}$. Also, Z represents a phase-flip error, where $Z|b\rangle = (-1)^b|b\rangle$.

In an n-qubit system, we usually use $|\psi\rangle \in \mathbb{C}^{2^n}$ to denote a quantum state. We can write $|\psi\rangle$ in the computational basis as $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$ where $\sum_x |\alpha_x|^2 = 1$. We identify $\mathbb{C}^{2^n}$ as $(\mathbb{C}^2)^{\otimes n}$ where each $\mathbb{C}_2$ denotes a qubit.

Suppose $|\psi\rangle$ is the correct codeword, and $|\psi'\rangle$ is the received codeword that might have corruptions. Our goal is to correct the possible errors in $|\psi'\rangle$ to recover $|\psi\rangle$. We recall the fundamental theorem in quantum ECC: to perform quantum error correction, it suffices to correct the following *discrete* set of error operators:

$$\mathcal{E} = \{E = E_1 \otimes E_2 \otimes \cdots \otimes E_n : E_i \in \{X, Z, I\}, |i : E_i \in \{X, Z\}| \le e\}. \tag{24.2}$$

The parameter $e \in \mathbb{N}$ is the number of qubit errors that the code can correct.

2 Detecting a Single Error

We start by considering the case of $e = 1$. The goal of this section is to develop methods to *detect* a single bit-flip error.

Let's first consider how to encode a single qubit. Classically, we have the repetition code $x \mapsto xxx$, which can detect a single bit-flip error. It also supports single-error correction by taking a majority vote over the received bits.

This inspires one to clone a qubit to solve the quantum version of the problem. Namely, one might want to design a unitary (an encoding operator) as $U : |\psi\rangle|0\rangle \mapsto |\psi\rangle|\psi\rangle$. Unfortunately, the famous quantum no-cloning theorem rules out the possibility of such a unitary.

Theorem 24.1 (Quantum no-cloning theorem)**.** *There is no unitary $U : \mathbb{C}^{2^n \times 2^n}$ that can satisfy $U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$ for all $\psi \in \mathbb{C}^{2^n}$.*

Proof. Suppose such a unitary U exists for $n = 1$ (the case of $n > 1$ is similar). For a state $|\psi\rangle \in \mathbb{C}^2$, write $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Then, by the "cloning" property of U, one has

$$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle = \alpha^2|00\rangle + \alpha\beta|01\rangle + \beta\alpha|10\rangle + \beta^2|11\rangle.$$

On the other hand, by the linearity of U, one has

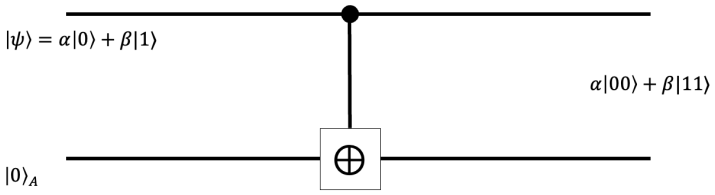
$$U|\psi\rangle|0\rangle = \alpha U|0\rangle|0\rangle + \beta U|1\rangle|1\rangle = \alpha|00\rangle + \beta|11\rangle.$$

This is clearly a contradiction. □

Theorem 24.1 shows that one cannot hope to copy an arbitrary quantum state. However, one can still design a unitary that faithfully copies a set of *orthogonal* states. In this section, we are interested in understanding how the unitary should act on each basis state in the computation basis (i.e., the basis $\{|x\rangle\}_{\{x\in\{0,1\}^k\}}$.

In particular, in the single-qubit case (i.e., $k = 1$), we want to design a unitary $U : \mathbb{C}^2 \times \mathbb{C}^{2^{n-1}} \to \mathbb{C}^{2^n}$ that maps $|b\rangle|0\rangle_{\text{ancilla}}$ to $|\psi_b\rangle$ for $b \in \{0, 1\}$. Here, $n - 1$ is the number of ancilla qubits we use, which also represents the redundancy of the code. The states $|\psi_b\rangle$ for $b \in \{0, 1\}$ (as well as their superpositions) are usually called *logical* qubits. Note that a logical qubit is implemented by n *physical* qubits.

Example 24.2. Consider a unitary U, which has $|0\rangle|0\rangle \mapsto |00\rangle$ and $|1\rangle|0\rangle \mapsto |11\rangle$. Intuitively, this unitary distributes the quantum information in the initial state across the entangled 2-qubit logic state. This unitary can be implemented by a CNOT gate. See the following figure.



We use $\mathcal{C} = U(\mathbb{C}^2 \times |0^{n-1}\rangle)$ to denote the space of the codeword. Note that $\mathcal{C} = \text{span}\{|\psi_0\rangle := |00\rangle, |\psi_1\rangle := |11\rangle\}$ is a 2-dimension subspace of $\mathbb{C}^4$.

Let $|\psi\rangle \in \mathcal{C}$ be a logic qubit. We can write $|\psi\rangle = \alpha|00\rangle + \beta|11\rangle$. Consider what happens if a single bit-flip error applies to $|\psi\rangle$. For example, consider

$$X_1|\psi\rangle = \alpha|10\rangle + \beta|01\rangle.$$

Here, X_1 denotes a bit-flip operator applied on the first qubit. We slightly abuse notation by using X_1 to refer to $X_1 \otimes I$. In general, we use a subscript below the operator to denote which qubit the operator acts on. We may concatenate several operators with different subscripts to denote their tensor product (e.g. $X_2Z_3 = X_2 \otimes Z_3$). We will frequently omit identity operators if the meaning is clear.

Continuing, we observe that

$$X_1|\psi\rangle \in \mathcal{F} := \text{span}\{|01\rangle, |10\rangle\}.$$

Intuitively, $\mathcal{F}$ is the error space. It is easy to see that $X_2|\psi\rangle \in \mathcal{F}$ as well.

Note that $\mathcal{F}$ is orthogonal to $\mathcal{C}$. Therefore, one can perform a suitable measurement to perfectly detect the error (without damaging valid logic qubits). In more detail, define two projections P_0, P_1, which project the states onto $\mathcal{C}$ and $\mathcal{F}$ respectively. One can see that $P_0P_1 = 0$ and $P_1 + P_2$. Then, given a received codeword $|\varphi\rangle$, performing the projection-valued measurement $\{P_0, P_1\}$ allows one to detect whether $|\phi\rangle \in \mathcal{F}$. Moreover, if $|\phi\rangle \in \mathcal{C}$, the measurement will not damage $|\phi\rangle$.

In fact, one can write down P_0 and P_1 explicitly:

$$P_0 = \frac{I \otimes I + Z_1 Z_2}{2}, \quad P_2 = \frac{I \otimes I - Z_1 Z_2}{2}. \tag{24.3}$$

3 Syndrome Extraction

We have shown a quantum code that can detect a single bit-flip error. Later we will generalize the construction and analysis to encode more qubits and detect more errors. Before that, it is helpful to develop a systematic method of syndrome (aka error) detection.

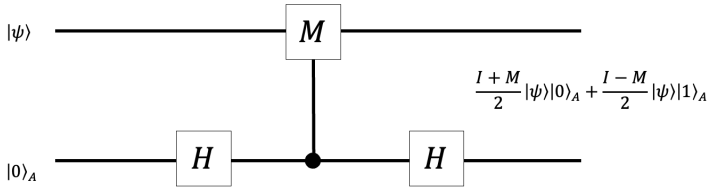
We illustrate the idea with Example 24.2. Let $|\psi\rangle \in \mathcal{C}$ be a logical qubit. Let $E|\psi\rangle$ be the received codeword where $E \in \{I, X_1, X_2\}$ denotes a possible error pattern. We want to distinguish between the case of $E = I$ from that $E = X_i$. We prepare one more ancilla bit to get $E|\psi\rangle|0\rangle_A$. Then we consider a unitary as

$$E|\psi\rangle|0\rangle_A \mapsto \frac{I_1 I_2 + Z_1 Z_2}{2} E|\psi\rangle|0\rangle_A + \frac{I_1 I_2 - Z_1 Z_2}{2} E|\psi\rangle|1\rangle_A. \tag{24.4}$$

It might not be obvious that this operation can be implemented by a unitary. Note that $Z_1 Z_2$ is a unitary with only ± 1 eigenvalues. We now prove that for any unitary M, we can implement an operator such that

$$|\psi\rangle|0\rangle_A \mapsto \frac{I + M}{2}|\psi\rangle|0\rangle_A + \frac{I - M}{2}|\psi\rangle|1\rangle_A. \tag{24.5}$$

In fact, this can be implemented by a controlled-M gate between two Hadamard gates on the ancilla bit, as shown in the following figure. It is straightforward to verify that the circuit implements the operation above faithfully.



Continuing, once we have the unitary as in (24.4), we can apply the operation and then measure the ancilla bit. It yields two possible outcomes. If $E = \{X_1, X_2\}$, then we get the result $|1\rangle_A$, which indicates that a bit-flip error happens. If $E \in \{I, X_1 X_2\}$ (i.e., if no error happens or the two qubits flip simultaneously), then we always get result $|0\rangle_A$. This shows that the measurement is capable of detecting a single bit-flip (but not two) error.

We call the outcome of the ancilla bit measurement an error "syndrome". In general, if we have a list of k "checks" $M_1, \ldots, M_k$, we can prepare one ancilla bit for each M_i, apply the operator of (24.5) for each M_i, and measure the ancilla bits in the computational basis. The outcome is called an error syndrome. The intuitive exposition here is formalized by the theory of stabilizer codes [Got97].

4 Correcting a Single Bit-Flip

We now use the syndrome extraction method to design a code that can *correct* a single-bit flip. The code in Example 24.2 cannot achieve this goal. We need a bit more redundancy as follows.

Example 24.3. Consider a unitary U, which satisfies that $|0\rangle|00\rangle \mapsto |000\rangle$ and $|1\rangle|00\rangle \mapsto |111\rangle$. Note that U can be implemented by two CNOT gates.

Given the code in Example 24.3, let $|\psi\rangle \in \mathcal{C} = \text{span}\{|000\rangle, |111\rangle\}$ be a logic qubit. We want to correct $X_1|\psi\rangle, X_2|\psi\rangle, X_3|\psi\rangle$. To achieve this goal, we need to detect which of X_1, X_2, X_3 happened without damaging the received state.

We observe that, as long as $|\psi\rangle \in \mathcal{C}$, we have

$$Z_1Z_2|\psi\rangle = |\psi\rangle, \quad Z_2Z_3|\psi\rangle = |\psi\rangle \tag{24.6}$$

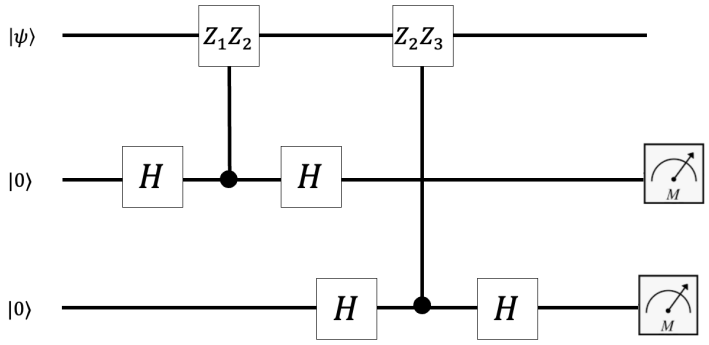
We call Z_1Z_2 and Z_2Z_3 the stabilizers of $\mathcal{C}$. We further observe that

$$Z_1Z_2(X_1|\psi\rangle) = -X_1|\psi\rangle.$$

Therefore, given a received codeword $E|\psi\rangle$ with the promise that $E \in \{I, X_1\}$, we can distinguish between the two cases of E by measuring with $M_1 = Z_1Z_2$ (similarly as what was done in Section 3). However, Z_1Z_2 cannot detect X_3. Namely,

$$Z_1Z_2(X_3|\psi\rangle) = X_3|\psi\rangle.$$

We observe that Z_2Z_3 can detect the error X_3. In fact, if we extract two syndromes with Z_1Z_2 and Z_2Z_3, then we are able to distinguish $E|\psi\rangle$ for $E \in \{I, X_1, X_2, X_3\}$. As we have discussed in Section 3, we can use two ancilla bits to perform the measurement. See the following figure for an illustration.



Based on the measurements of the ancilla bits, we can distinguish different types of errors, as shown in the following table.

type of error	Z_1Z_2	Z_2Z_3
I	0	0
X_1	1	0
X_2	1	1
X_3	1	1

Finally, once we know which type of error happens, we can correct the error by applying the appropriate operator. For example, if we know that the received codeword is $X_1|\psi\rangle$. We can apply X_1 again to get $(X_1X_1)|\psi\rangle = |\psi\rangle$.

Note that the role of Z_1Z_2 and Z_2Z_3 is similar to the parity check matrix in the classical coding theory. We can generalize the argument to handle more bit-flip errors. Given a quantum code $\mathcal{C} \subseteq \mathbb{C}^{2^n}$, consider an error pattern $X_S = \prod_{i\in S} X_i$, and a stabilizer $Z_T = \prod_{i\in T} Z_i$ for $\mathcal{C}$. Note that Z_T can detect X_S, if and only if $|S \cap T| \equiv 1 \bmod 2$. Therefore, the quantum code fails to detect an error pattern X_S, if and only if every stabilizer Z_T of $\mathcal{C}$ has an even intersection with X_S (i.e., $|S \cap T| \equiv 0 \bmod 2$).

In fact, one can take the parity check matrix of any linear code to correct (quantum) bit-flip errors up to the code's (classical) error correction capacity. Namely, we can use the parity check matrix H of any linear code $C : \mathbb{F}_2^k \to \mathbb{F}_k^n$ of distance d to construct a quantum code. In this way, we get a quantum code that can correct up to $\frac{d}{2}$ X errors.

For example, we can convert the Hamming code ($[7, 4, 3]_{\mathbb{F}_2}$-code) to a quantum code $\mathcal{C} \subseteq \mathbb{C}^{2^7}$ of dimension 4, that is capable of correcting any single bit-flip error. Recall the parity check matrix of the Hamming code

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

The stabilizers for $\mathcal{C}$ will be $\{Z_4Z_5Z_6Z_7, Z_2Z_3Z_6Z_7, Z_1Z_3Z_5Z_7\}$.

5 Shor's Code

So far we have discussed how to correct bit-flip errors. Now we discuss how to correct *both* bit-flip and error-flip errors. Recall the 3-bit-repetition code in Example 24.3 as $C_{3b} = \text{span}\{|000\rangle, |111\rangle\}$. We have shown this code can correct a single bit-flip. However, it fails to even *detect* a single phase-flip error (say, Z_1).

Consider the $|\pm\rangle$ basis

$$|+\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{24.7}$$

Note that phase-flip errors are bit-flip errors in the $|\pm\rangle$ basis. By doing repetition in the $|\pm\rangle$ basis, we can construct a three-phase-repetition code $C_{3p} = \text{span}\{|{+}{+}{+}\rangle, |{-}{-}{-}\rangle\}$ to correct a single phase-flip code.

Shor [Sho95] designed a code that can correct both bit-flip and phase-flip errors. The code was constructed by contactenting C_{3p} with C_{3b}.

Namely, given a quantum qubit $|b\rangle$, we first use C_{3p} to encode it as $|{+}{+}{+}\rangle$ or $|{-}{-}{-}\rangle$. Then we apply C_{3b} on each of the 3 qubits. Observe that this construction gives

$$|0\rangle|0^8\rangle \mapsto \frac{|000\rangle + |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle + |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle + |111\rangle}{\sqrt{2}}$$

and

$$|1\rangle|0^8\rangle \mapsto \frac{|000\rangle - |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle - |111\rangle}{\sqrt{2}} \otimes \frac{|000\rangle - |111\rangle}{\sqrt{2}}.$$

In this way, we get Shor's code $\mathcal{C}_{shor} \subseteq \mathbb{C}^{2^9}$ where

$$\mathcal{C}_{shor} = \left\{ |0\rangle_{shor} = \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}} \right)^{\otimes 3}, |1\rangle_{shor} = \left(\frac{|000\rangle - |111\rangle}{\sqrt{2}} \right)^{\otimes 3} \right\}. \tag{24.8}$$

We can list 8 stabilizers for $\mathcal{C}_{shor}$.

$$\begin{aligned} &Z_1Z_2, \quad Z_2Z_3, \\ &Z_4Z_5, \quad Z_5Z_6, \\ &Z_7Z_8, \quad Z_7Z_8, \\ &X_1X_2X_3X_4X_5X_6, \\ &X_4X_5X_6X_7X_8X_9. \end{aligned}$$

Call them $H_1, \ldots, H_8$. Then, one can verify that $\mathcal{C}_{shor} = \{|\psi\rangle : H_i|\psi\rangle = |\psi\rangle\}$.

We can similarly study how these stabilizers detect different bit-flip or phase-flip errors. Interestingly, there is a degenerate phenomenon. Namely, one cannot distinguish among Z_1, Z_2, Z_3. For any $|\psi\rangle \in \mathcal{C}_{shor}$, it holds that

$$Z_1|\psi\rangle = Z_2|\psi\rangle = Z_3|\psi\rangle.$$

Nevertheless, this is not an issue for us. Because we can correct all of them by applying Z_1.

In the next lecture, we will formally study how to use Shor's code to correct an arbitrary single-qubit error (either bit-flip or phase-flip).

References

[Got97] Daniel Gottesman. Stabilizer codes and quantum error correction, 1997. 3

[Pre] John Preskill. Chapter 7, lecture notes for quantum computation. 1

[Rof19] Joschka Roffe. Quantum error correction: an introductory guide. *Contemporary Physics*, 60(3):226–245, jul 2019. 1

[Sho95] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995. 5