TWELVE LECTURES ON QUANTUM ERROR CORRECTION

SAYAN CHAKRABORTY

Contents

1. Lecture 1 & 2: quantum states, channels, and the postulates of quantum mechanics

Notation 1.1. $\mathcal{H}$ will always denote a finite-dimensional Hilbert space with the inner product $\langle \phi, \psi \rangle$ for $\phi, \psi \in \mathcal{H}$. Hence, $\mathcal{H}$ is isomorphic to $\mathbb{C}^n$ for some n, which is the dimension of the Hilbert space $\mathcal{H}$. We will use the convention that the inner product is linear in the second component.

Notation 1.2. For two Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, $\mathcal{B}(\mathcal{H}_1, \mathcal{H}_2)$ will denote the set of all linear maps from $\mathcal{H}_1$ to $\mathcal{H}_2$. The space $\mathcal{B}(\mathcal{H}, \mathcal{H})$ will simply be denoted by $\mathcal{B}(\mathcal{H})$.

Notation 1.3. For a map $T \in \mathcal{B}(\mathcal{H}_1, \mathcal{H}_2)$, $T^\dagger$ will denote the Hermitian conjugate of T, which lies in $\mathcal{B}(\mathcal{H}_2, \mathcal{H}_1)$. Hence, if $T \in \mathcal{B}(\mathcal{H})$, then $T^\dagger \in \mathcal{B}(\mathcal{H})$ as well. The identity operator in $\mathcal{B}(\mathcal{H})$ will be denoted by I.

Notation 1.4. For $\psi \in \mathcal{H}$, if the norm of ψ satisfies $\|\psi\| = 1$, then we will denote the vector ψ by $|\psi\rangle$ (called "ket ψ"). We will sometimes abuse notation by writing $|\psi\rangle$ for an arbitrary vector ψ. Note that we can view $|\psi\rangle$ as an element of $\mathcal{B}(\mathbb{C}, \mathcal{H})$, acting by multiplication. Then $(|\psi\rangle)^\dagger$ is an element of $\mathcal{B}(\mathcal{H}, \mathbb{C})$, which we denote by $\langle \psi|$ (called "bra ψ"). For any $|\phi\rangle \in \mathcal{H}$, the expression $\langle \psi| \, (|\phi\rangle)$ will be written using the "braket" notation as $\langle \psi | \phi \rangle$. Note that $\langle \psi | \phi \rangle$ is simply the inner product $\langle \psi, \phi \rangle$.

Notation 1.5. The standard basis of $\mathcal{H} = \mathbb{C}^n$, $\{e_i\}$ for $i = 1, 2, \ldots, n$, where e_i denotes the vector with 1 in the i-th component and zero elsewhere, will be denoted by the vectors $\{|i\rangle\}$ for $i = 0, 1, \ldots, n-1$. Here, $|i\rangle = e_{i+1}$.

Notation 1.6. For two Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, $\mathcal{H}_1 \otimes \mathcal{H}_2$ will denote their tensor product. The n-fold tensor product of a Hilbert space $\mathcal{H}$ will be denoted by $\mathcal{H}^{\otimes n}$.

To denote $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle \in \mathcal{H}^{\otimes n}$, we will often use the shorthand notation $|\psi_1\rangle |\psi_2\rangle \cdots |\psi_n\rangle$ or $|\psi_1 \psi_2 \cdots \psi_n\rangle$.

This notation is convenient for the following reason: if $\mathcal{H} = \mathbb{C}^2$ with standard basis $|0\rangle$ and $|1\rangle$, then the standard basis of $\mathcal{H}^{\otimes 2}$ will be $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ under our notation. When identifying $\mathcal{H}^{\otimes 2}$ with $\mathbb{C}^4$, $|00\rangle$ corresponds to $|0\rangle$, $|01\rangle$ to $|1\rangle$, $|10\rangle$ to $|2\rangle$, and $|11\rangle$ to $|3\rangle$, where $\{|i\rangle\}$ for $i = 0, 1, 2, 3$ is the standard basis of $\mathbb{C}^4$. Note that 00, 01, 10, and 11 are simply the binary representations of 0, 1, 2, and 3, respectively. The same idea applies to $\mathcal{H}^{\otimes n}$ for any n.

Notation 1.7. For $z \in \mathbb{C}$, the complex conjugate of z will be denoted by z^*.

Exercise 1.8. ★ [1] *Let $\{\phi_k\}_{k=1}^d$ and $\{\psi_k\}_{k=1}^d$ be sets of vectors in a Hilbert space $\mathcal{H}$. Assume that*

$$\sum_{k=1}^{d} \phi_k \otimes \psi_k = 0.$$

Show that if the vectors $\{\psi_k\}$ are linearly independent, then $\phi_k = 0$ for all k.

We begin with a basic yet fundamental result in quantum information theory.

[1] ★ = easy, ★ = medium, ★ = slightly hard

Theorem 1.9 (No-Cloning)**.** *Let $\mathcal{H} = \mathbb{C}^n$ and let $|\theta\rangle$ be a fixed vector in $\mathcal{H}$. Then there does not exist an operator $U \in \mathcal{B}(\mathcal{H} \otimes \mathcal{H})$ such that*

$$U(|\psi\rangle \otimes |\theta\rangle) = |\psi\rangle \otimes |\psi\rangle\,, \quad \text{for all } |\psi\rangle \in \mathcal{H}.$$

Proof. This follows immediately from the linearity of U. Assume such an operator U exists. Then, by hypothesis,

$$U(|0\rangle \otimes |\theta\rangle) = |0\rangle \otimes |0\rangle\,,$$
$$U(|1\rangle \otimes |\theta\rangle) = |1\rangle \otimes |1\rangle\,.$$

For any vector of the form $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$, linearity of U gives

$$U(|\psi\rangle \otimes |\theta\rangle) = a\,|0\rangle \otimes |0\rangle + b\,|1\rangle \otimes |1\rangle\,. \tag{1.1}$$

On the other hand, we also have

$$U(|\psi\rangle \otimes |\theta\rangle) = |\psi\rangle \otimes |\psi\rangle = a^2\,|0\rangle \otimes |0\rangle + ab\,|0\rangle \otimes |1\rangle + ab\,|1\rangle \otimes |0\rangle + b^2\,|1\rangle \otimes |1\rangle\,. \tag{1.2}$$

The expressions in (1.1) and (1.2) differ in general. For instance, if $a = -1$ and $b = 0$, equality of the right-hand sides would imply that $|0\rangle \otimes |0\rangle$ is the zero vector, which is a contradiction. Hence, no such operator U exists. □

Exercise 1.10. ★ *Let $\mathcal{H}$ be a Hilbert space, and let $|\phi\rangle, |\eta\rangle, |\eta'\rangle$ be fixed unit vectors in $\mathcal{H}$. Show that there does not exist an operator $U \in \mathcal{B}(\mathcal{H}^{\otimes 3})$ such that*

$$U(|\psi\rangle \otimes |\psi\rangle \otimes |\eta\rangle) = |\psi\rangle \otimes |\phi\rangle \otimes |\eta'\rangle\,, \quad \text{for all } |\psi\rangle \in \mathcal{H}.$$

1.1. **Postulates of Quantum Mechanics.** In this section, we discuss the fundamental postulates of quantum mechanics.

Postulate 1: Any closed physical system A is described by a complex Hilbert space $\mathcal{H}$, known as the *state space* of the system. A *state* of the system is represented by a unit vector $|\psi\rangle \in \mathcal{H}$.

Notation 1.11. We say that a linear combination

$$|\psi\rangle := \sum_i \alpha_i\,|\psi_i\rangle$$

is a *superposition* of the states $\{|\psi_i\rangle\}$. For $|\psi\rangle$ to represent a valid quantum state, the *amplitudes* α_i must satisfy

$$\sum_i |\alpha_i|^2 = 1.$$

For example, the following are superposition states:

$$|+\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad |-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Postulate 2: The *evolution* of a state in a closed quantum system with state space $\mathcal{H}$ is described by a unitary transformation $U \in \mathcal{B}(\mathcal{H})$. More precisely, if the state of the system at time t_1 is $|\psi(t_1)\rangle$ and at time t_2 is $|\psi(t_2)\rangle$, then

$$|\psi(t_2)\rangle = U\,|\psi(t_1)\rangle\,,$$

where $U = U_{t_1,t_2} \in \mathcal{B}(\mathcal{H})$ is a unitary operator that depends on t_1 and t_2.

Postulate 3 (Measurement Postulate): *Quantum measurements* are described by a finite collection of operators M_m, called *measurement operators*, acting on the Hilbert space $\mathcal{H}$ of the system. These operators satisfy the *completeness relation*:

$$\sum_m M_m^\dagger M_m = \mathrm{I}.$$

When a *measurement* described by $\{M_m\}$ is performed on a state $|\psi\rangle$, the outcome m occurs with probability

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

and, conditioned on this outcome, the *post-measurement state* of the system is

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

The completeness relation ensures that the probabilities sum to one:

$$\sum_m p(m) = 1.$$

The above description is the general measurement postulate. We now discuss two important cases: *projective measurement* and *POVM measurement*.

Projective Measurement. A projective measurement is described by a single observable M, which is a Hermitian operator acting on the Hilbert space $\mathcal{H}$. The spectral decomposition of M is

$$M = \sum_m m P_m,$$

where P_m is the projection operator onto the eigenspace corresponding to the eigenvalue m. Since

$$\mathrm{I} = \sum_m P_m,$$

we can take $\{P_m\}$ as the measurement operators. In this case, the possible measurement outcomes are the eigenvalues of M. The probability of obtaining outcome m is

$$p(m) = \langle\psi| P_m |\psi\rangle ,$$

and the post-measurement state becomes

$$\frac{P_m |\psi\rangle}{\sqrt{p(m)}}.$$

An important property of projective measurements is that the expected value of the observable M can be expressed concisely as

$$\begin{aligned}\mathbb{E}(M) &= \sum_m m\, p(m) \\ &= \sum_m m\ \langle\psi| P_m |\psi\rangle \\ &= \langle\psi| M |\psi\rangle .\end{aligned}$$

POVM Measurement. POVM stands for *Positive Operator-Valued Measure*. In this case, instead of the measurement operators $\{M_m\}$, we are given a collection of positive operators $\{E_m\}$ satisfying

$$\sum_m E_m = \mathrm{I}.$$

The set $\{E_m\}$ is called a POVM. For each E_m, we can choose a positive square root M_m such that $E_m = M_m^\dagger M_m$. Hence, any POVM corresponds to a family of measurement operators, and there is a one-to-one correspondence between $\{E_m\}$ and such families $\{M_m\}$.

Postulate 4: Suppose we have two closed physical systems, A and B, described by the Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively. Then the joint system AB is described by the tensor product Hilbert space $\mathcal{H}_1 \otimes \mathcal{H}_2$.

We now introduce the concept of a *mixed state*, which generalizes the notion of a quantum state. Instead of representing a state as a vector $|\psi\rangle$ in a Hilbert space $\mathcal{H}$, we represent it by the rank-one projection operator $|\psi\rangle\langle\psi|$ in $\mathcal{B}(\mathcal{H})$. The operator $|\psi\rangle\langle\psi|$ *remembers* all the information of the state $|\psi\rangle$.

In this formalism, a (mixed) state ρ is defined as

$$\rho := \sum_i p_i |\psi_i\rangle\langle\psi_i|, \tag{1.3}$$

where $\{|\psi_i\rangle\}$ is a collection of state vectors and $\{p_i\}$ is a set of non-negative real numbers satisfying $\sum_i p_i = 1$.

Note that ρ is a positive operator with trace equal to 1. In fact, by the spectral theorem, any positive operator of trace 1 can be expressed in the form

$$\rho = \sum_{i=1}^{n} p_i |\psi_i\rangle\langle\psi_i|.$$

Therefore, we define a general quantum state as a positive operator on $\mathcal{H}$ with trace 1. Such an operator is called a *density operator*. When the state is of the special form $\rho = |\psi\rangle\langle\psi|$, it is called a *pure state*; otherwise, it is referred to as a *mixed state*.

With this description in hand, we can now restate the postulates of quantum mechanics in terms of density operators.

Postulate 1: Any closed physical system is described by a complex Hilbert space $\mathcal{H}$. A state of the system is represented by a density operator $\rho \in \mathcal{B}(\mathcal{H})$, which is a positive operator with trace equal to 1.

Postulate 2: The evolution of a state in a closed quantum system with state space $\mathcal{H}$ is described by a unitary transformation U. If the state of the system at time t_1 is ρ_1 and at time t_2 is ρ_2, then

$$\rho_2 = U\rho_1 U^\dagger,$$

where $U = U_{t_1,t_2} \in \mathcal{B}(\mathcal{H})$ is a unitary operator that depends on t_1 and t_2.

Postulate 3: Quantum measurements are described by a collection of operators

$$\{M_m\},$$

called *measurement operators*, acting on the Hilbert space $\mathcal{H}$ of the system. These operators satisfy the *completeness relation*:

$$\sum_m M_m^\dagger M_m = \mathrm{I}.$$

When a measurement is performed on a state ρ, the probability of obtaining outcome m is

$$p(m) = \mathrm{Tr}(M_m^\dagger M_m \rho),$$

and the post-measurement state (conditioned on outcome m) is

$$\rho' = \frac{M_m \rho M_m^\dagger}{\mathrm{Tr}(M_m^\dagger M_m \rho)}.$$

Postulate 4: Remains the same as before: if two systems are described by Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, then the joint system is described by the tensor product Hilbert space $\mathcal{H}_1 \otimes \mathcal{H}_2$.

Exercise 1.12. ★ *Show that a state ρ satisfies $\mathrm{Tr}(\rho^2) \leq 1$ and that it is mixed if and only if $\mathrm{Tr}(\rho^2) < 1$.*

Exercise 1.13. ★ *Let $\{|\psi_i\rangle\}$ and $\{|\varphi_j\rangle\}$ be two sets of vectors in a Hilbert space $\mathcal{H}$ (add zeros if necessary so that both sets have the same number of elements). Show that the operators*

$$\sum_i |\psi_i\rangle \langle\psi_i| \quad \text{and} \quad \sum_j |\varphi_j\rangle \langle\varphi_j|$$

are equal if and only if

$$|\psi_i\rangle = \sum_j u_{ij} |\varphi_j\rangle ,$$

where (u_{ij}) is a unitary matrix.

1.2. **Quantum Entanglement.** As we have seen, a composite quantum system is modeled by the tensor product of Hilbert spaces. Suppose we have (possibly mixed) states $\rho_i \in \mathcal{B}(\mathcal{H}_i)$, for $1 \leq i \leq n$. Then a canonical state of the composite system is given by

$$\rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_n \in \mathcal{B}(\mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_n).$$

A quantum state $\rho \in \mathcal{B}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ is said to be *separable* if and only if there exist states $\rho_{1,j} \in \mathcal{B}(\mathcal{H}_1)$ and $\rho_{2,j} \in \mathcal{B}(\mathcal{H}_2)$, with $j \in \{1, \ldots, m\}$ and probabilities p_j satisfying $\sum_{j=1}^m p_j = 1$, such that

$$\rho = \sum_{j=1}^m p_j (\rho_{1,j} \otimes \rho_{2,j}).$$

A quantum state $\rho \in \mathcal{B}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ is called *entangled* if it is not separable.

Exercise 1.14. ★ *Show that a pure state $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ is separable if and only if it is the tensor product of two pure states, i.e.,*

$$|\psi\rangle = |\psi\rangle_1 \otimes |\psi\rangle_2,$$

where $|\psi\rangle_1 \in \mathcal{H}_1$ and $|\psi\rangle_2 \in \mathcal{H}_2$. (Hint: use Exercise 1.12.)

To give some examples, it can be checked that

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad \text{and} \quad \left|\Phi^-\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

are entangled states. These are known as *Bell states*. In general, for $\mathcal{H} = \mathbb{C}^n$, the state

$$|\mathrm{Bell}\rangle = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |ii\rangle \in \mathcal{H}^{\otimes 2}$$

is also an entangled state. For $0 \leq i, j \leq n-1$, we denote the *matrix units* by

$$e(i,j) = |i\rangle\langle j|,$$

where $\{|i\rangle\}_{i=0}^{n-1}$ represents the standard basis for $\mathcal{H} = \mathbb{C}^n$. It is clear that $\{e(i,j)\}$ forms a basis of

$$\mathcal{B}(\mathcal{H}) = \mathrm{M}_n(\mathbb{C}).$$

Exercise 1.15. ★ *Consider the matrix of matrix units $(e(i,j)) \in \mathrm{M}_n(\mathcal{B}(\mathcal{H}))$. Show that*

$$(e(i,j)) = |\mathrm{Bell}\rangle\langle\mathrm{Bell}|.$$

1.3. **Open quantum systems and quantum channels.** In the formalism of *open quantum systems*, the unitary evolution described in Postulate 2 is replaced by a more general notion: evolution by a *quantum channel*. To keep the discussion mathematically focused, we avoid relying too heavily on physical intuition.

Suppose we have a quantum system associated with a Hilbert space $\mathcal{H}$, which can be decomposed as a tensor product

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2,$$

where $\mathcal{H}_1$ and $\mathcal{H}_2$ represent two subsystems. If the overall system undergoes unitary evolution on the full space $\mathcal{H}$, and we are interested only in the dynamics of the subsystem $\mathcal{H}_1$, then the effective evolution on $\mathcal{H}_1$ is, in general, no longer unitary. Instead, it is described by a quantum channel—which we will be describing below.

This formalism captures the idea that when part of a larger system is ignored (or "traced out"), the remaining part evolves in a way that can incorporate decoherence and noise, phenomena that cannot be described by unitary evolution alone.

Definition 1.16. *Let $\mathcal{H}$ be an n-dimensional Hilbert space. A* quantum channel *$\mathcal{E}$ is a completely positive linear map from $\mathcal{B}(\mathcal{H})$ to itself satisfying*

$$0 \leq \mathrm{Tr}(\mathcal{E}(\rho)) \leq \mathrm{Tr}(\rho), \quad \textit{for all positive } \rho \in \mathcal{B}(\mathcal{H}).$$

A *trace-preserving quantum channel* $\mathcal{E}$ is a completely positive linear map from $\mathcal{B}(\mathcal{H})$ to itself such that

$$\mathrm{Tr}(\mathcal{E}(\rho)) = \mathrm{Tr}(\rho), \quad \text{for all } \rho \in \mathcal{B}(\mathcal{H}).$$

The following theorem describes the structure of quantum channels.

Theorem 1.17 (Kraus Representation)**.** *Let $\mathcal{E}$ be a completely positive map from $\mathcal{B}(\mathcal{H})$ to itself, where $\mathcal{H}$ is an n-dimensional Hilbert space. Then there exist operators $\{E_k\}_{k=1}^{n^2} \subseteq \mathcal{B}(\mathcal{H})$ such that*

$$\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger, \quad \textit{for all } \rho \in \mathcal{B}(\mathcal{H}).$$

Moreover:

- *If $\mathcal{E}$ is a quantum channel, then $\sum_k E_k^\dagger E_k \leq \mathrm{I}$.*
- *If $\mathcal{E}$ is a trace-preserving quantum channel, then $\sum_k E_k^\dagger E_k = \mathrm{I}$.*

Proof. Denote by R the $n^2 \times n^2$ positive matrix

$$R := \mathcal{E}^{(n)}\big((e(i,j))\big),$$

where $\mathcal{E}^{(n)}$ denotes the *amplification* of $\mathcal{E}$ to $\mathrm{M}_n(\mathcal{B}(\mathcal{H}))$: $\mathcal{E}^{(n)} = \mathcal{E} \otimes \mathrm{I}$ as a map from $\mathrm{M}_n(\mathcal{B}(\mathcal{H})) = \mathcal{B}(\mathcal{H}) \otimes \mathrm{M}_n(\mathbb{C})$ to itself. The positivity of R follows from the complete positivity of $\mathcal{E}$ and the fact that $(e(i,j))$ is a positive element (see Exercise 1.15).

Using the spectral decomposition of R, we write

$$R = \sum_{k=1}^{r} |\psi_k\rangle \langle\psi_k|,$$

where $|\psi_k\rangle \in \mathbb{C}^{n^2}$ are eigenvectors of R, and $r = \mathrm{rank}(R)$.
Next, observe that we can view $\mathbb{C}^{n^2}$ as the direct sum of n copies of $\mathbb{C}^n$, indexed by the first tensor component.

Let $\{P_i : 1 \leq i \leq n\}$ denote the family of rank-n projections, where each P_i is the $n \times n^2$ matrix that projects onto the i-th copy of $\mathbb{C}^n$ in the direct sum decomposition of $\mathbb{C}^{n^2}$. Equivalently, P_i extracts the block corresponding to the i-th summand, and $P_i^\dagger$ is the $n^2 \times n$ inclusion map embedding $\mathbb{C}^n$ into the i-th block.

These projections satisfy

$$P_i R P_j^\dagger = \mathcal{E}(e(i,j)) \quad \text{and} \quad |\psi_k\rangle = \sum_{i=1}^{n} P_i^\dagger P_i \, |\psi_k\rangle \, .$$

For $1 \leq k \leq r$, define a linear operator $E_k : \mathbb{C}^n \to \mathbb{C}^n$ by

$$E_k \, |i\rangle := P_i \, |\psi_k\rangle \, , \quad 1 \leq i \leq n.$$

That is, the i-th column of E_k is given by $P_i \, |\psi_k\rangle$.
Now compute:

$$\begin{aligned} R &= \sum_k \sum_{i,j} P_i^\dagger P_i \, |\psi_k\rangle \, \langle\psi_k| \, P_j^\dagger P_j \\ &= \sum_{i,j} P_i^\dagger \Big(\sum_k E_k \, |i\rangle \, \langle j| \, E_k^\dagger \Big) P_j. \end{aligned}$$

Hence, for the matrix units $e(i,j) = |i\rangle \, \langle j|$,

$$\mathcal{E}(e(i,j)) = \mathcal{E}(|i\rangle \, \langle j|) = P_i R P_j^\dagger = \sum_{k=1}^{r} E_k \, |i\rangle \, \langle j| \, E_k^\dagger, \tag{1.4}$$

using the fact that $P_i P_j^\dagger = \delta_{ij} I$.

Since $\{e(i,j)\}$ forms a basis of $\mathcal{B}(\mathbb{C}^n)$, Equation (1.4) extends to all $\rho \in \mathcal{B}(\mathbb{C}^n)$, completing the proof.

Now assume that $\mathcal{E}$ satisfies

$$\mathrm{Tr}(\mathcal{E}(\rho)) \leq \mathrm{Tr}(\rho), \quad \text{for all positive } \rho \in \mathcal{B}(\mathcal{H}).$$

Define

$$E := \sum_k E_k^\dagger E_k.$$

Clearly, E is a Hermitian operator. By the spectral theorem, we can write

$$E = \sum_k \lambda_k \, |\phi_k\rangle \, \langle\phi_k| \, ,$$

where $\{|\phi_k\rangle\}$ is an orthonormal basis of $\mathcal{H}$ and $\lambda_k \in \mathbb{R}$. It is enough to show that $\mathrm{I} - E$ is positive.
To this end, take any vector $|\psi\rangle \in \mathcal{H}$. Then

$$\langle\psi| \, (\mathrm{I} - E) \, |\psi\rangle = \mathrm{Tr} \left((\mathrm{I} - E) \, |\psi\rangle \, \langle\psi| \, \right).$$

Set $\rho := |\psi\rangle \, \langle\psi|$, which is positive with $\mathrm{Tr}(\rho) = 1$. Then

$$\mathrm{Tr} \left((\mathrm{I} - E)\rho \right) = \mathrm{Tr}(\mathrm{I}\rho) - \mathrm{Tr}(E\rho) = \mathrm{Tr}(\rho) - \mathrm{Tr}(E\rho).$$

Note that

$$\mathrm{Tr}(E\rho) = \mathrm{Tr} \left(\sum_k E_k^\dagger E_k \rho \right) = \mathrm{Tr} \left(\sum_k E_k \rho E_k^\dagger \right) = \mathrm{Tr}(\mathcal{E}(\rho)).$$

Thus

$$\mathrm{Tr} \left((I - E)\rho \right) = \mathrm{Tr}(\rho) - \mathrm{Tr}(\mathcal{E}(\rho)) \geq 0,$$

by assumption. Since this holds for all $|\psi\rangle$, it follows that $\mathrm{I} - E \geq 0$, i.e., $\mathrm{I} - E$ is positive. Now assume that $\mathcal{E}$ is trace-preserving. Let ρ be any operator with $\mathrm{Tr}(\rho) = 1$. Then

$$1 = \mathrm{Tr}(\mathcal{E}(\rho)) = \mathrm{Tr}\left(\sum_k E_k \rho E_k^\dagger\right) = \mathrm{Tr}\left(\sum_k E_k^\dagger E_k \rho\right) = \mathrm{Tr}(E\rho).$$

Since this equality holds for all ρ with $\mathrm{Tr}(\rho) = 1$, it must be that

$$E = \sum_k E_k^\dagger E_k = \mathrm{I}.$$

Indeed, write $E = (E_{ij})$ in the standard basis. We verify that $E_{ij} = \delta_{ij}$. For $i = j$, take the matrix unit $e(i,i)$, which satisfies $\mathrm{Tr}(e(i,i)) = 1$. Then

$$\mathrm{Tr}(Ee(i,i)) = E_{ii} = 1.$$

For $i \neq j$, consider $e = e(j,i) + e(1,1)$, which has $\mathrm{Tr}(e) = 1$. Then

$$\mathrm{Tr}(Ee) = E_{ij} + 1.$$

Since $\mathrm{Tr}(Ee) = \mathrm{Tr}(e) = 1$, it follows that $E_{ij} = 0$. Therefore, $E = \mathrm{I}$, as required.

□

The converse of the previous theorem is easier to verify and is stated below.

Theorem 1.18. *Let $\mathcal{E}$ be a map from $\mathcal{B}(\mathcal{H})$ to itself such that there exist operators $\{E_k\}_{k=1}^r$ satisfying $\sum_{k=1}^r E_k^\dagger E_k \leq \mathrm{I}$, and*

$$\mathcal{E}(\rho) = \sum_{k=1}^r E_k \rho E_k^\dagger, \quad \textit{for all } \rho \in \mathcal{B}(\mathcal{H}),$$

for some integer r. Then $\mathcal{E}$ is a quantum channel. Moreover, if $\sum_{k=1}^r E_k^\dagger E_k = \mathrm{I}$, then $\mathrm{Tr}(\mathcal{E}(\rho)) = \mathrm{Tr}(\rho)$ for all positive $\rho \in \mathcal{B}(\mathcal{H})$.

Proof. The last assertion follows immediately:

$$\mathrm{Tr}(\mathcal{E}(\rho)) = \mathrm{Tr}\left(\sum_k E_k \rho E_k^\dagger\right) = \mathrm{Tr}\left(\sum_k E_k^\dagger E_k \rho\right) = \mathrm{Tr}(\mathrm{I}\rho) = \mathrm{Tr}(\rho).$$

Assume $E := \sum_k E_k^\dagger E_k \leq \mathrm{I}$. We first show that $0 \leq \mathrm{Tr}(\mathcal{E}(\rho)) \leq \mathrm{Tr}(\rho)$ for all positive $\rho \in \mathcal{B}(\mathcal{H})$. Define $S := \mathrm{I} - E$, which is positive. Let the spectral decomposition of S be

$$S = \sum_i \lambda_i |\phi_i\rangle \langle\phi_i|, \quad \lambda_i \geq 0.$$

Then, for any positive ρ,

$$\mathrm{Tr}(S\rho) = \mathrm{Tr}\left(\sum_i \lambda_i |\phi_i\rangle \langle\phi_i| \rho\right) = \sum_i \lambda_i \langle\phi_i| \rho |\phi_i\rangle \geq 0.$$

Thus $\mathrm{Tr}((\mathrm{I}-E)\rho) \geq 0$, implying $\mathrm{Tr}(E\rho) \leq \mathrm{Tr}(\rho)$. Since $\mathrm{Tr}(E\rho) = \mathrm{Tr}(\mathcal{E}(\rho))$, we have $\mathrm{Tr}(\mathcal{E}(\rho)) \leq \mathrm{Tr}(\rho)$. Clearly, $\mathrm{Tr}(\mathcal{E}(\rho)) \geq 0$ once we show that $\mathcal{E}(\rho)$ is positive.

Next, we show that $\mathcal{E}$ is completely positive. Let $\mathcal{E}^{(l)} := \mathcal{E} \otimes \mathrm{I}$ denote the map from $\mathcal{B}(\mathcal{H}) \otimes \mathrm{M}_l(\mathbb{C})$ to itself. We need to show that $(\mathcal{E} \otimes I)(\Phi)$ is positive for all positive Φ acting on $\mathcal{H} \otimes \mathbb{C}^l$. For any vector $|\psi\rangle \in \mathcal{H} \otimes \mathbb{C}^l$,

$$\left\langle \psi \middle| \left((E_i \otimes \mathrm{I})\Phi(E_i^\dagger \otimes \mathrm{I})\right) \middle| \psi \right\rangle \geq 0,$$

since Φ is positive. Therefore,

$$\langle\psi|(\mathcal{E}\otimes \mathrm{I})(\Phi)|\psi\rangle = \sum_i \left\langle \psi \middle| (E_i\otimes \mathrm{I})\Phi(E_i^\dagger\otimes \mathrm{I}) \middle| \psi \right\rangle \geq 0.$$

The equality holds because any element of the tensor product can be expressed as a linear combination of elementary tensors. Hence $(\mathcal{E}\otimes \mathrm{I})(\Phi)$ is positive, which proves that $\mathcal{E}$ is completely positive. □

Theorems 1.17 and 1.18 provide a characterization of a quantum channel in terms of a set of operators $\{E_k\}_{k=1}^r$. This collection $\{E_k\}_{k=1}^r$ is referred to as the Kraus operators, also known as noise operators (or errors) of the channel $\mathcal{E}$. It is important to note that the set of Kraus operators is not unique. In fact, we have the following proposition.

Proposition 1.19. *Let $\{E_1,\dots,E_r\}$ and $\{E'_1,\dots,E'_s\}$ be the Kraus operators corresponding to the completely positive maps $\mathcal{E}$ and $\mathcal{E}'$, respectively, as in Theorem 1.17. Assume that $s=r$ (which can always be arranged by adding zero operators if necessary). Then $\mathcal{E}=\mathcal{E}'$ if and only if there exists an $r\times r$ complex unitary matrix $U=(u_{ij})$ such that*

$$E_i = \sum_{j=1}^r u_{ij}E'_j, \quad \textit{for all } 1\leq i\leq r.$$

Proof. Suppose $\{E_i\}$ and $\{E'_i\}$ are two sets of Kraus operators corresponding to the same completely positive map, i.e.,

$$\mathcal{E}(\rho) = \sum_i E_i\rho E_i^\dagger = \sum_j E'_j\rho {E'_j}^\dagger = \mathcal{E}'(\rho) \quad \text{for all } \rho.$$

Define the following vectors on $\mathcal{H}\otimes\mathcal{H}$:

$$|e_i\rangle := \sum_k |k\rangle\otimes\left(E_i|k\rangle\right) = \sum_k (\mathrm{I}\otimes E_i)(|k\rangle\otimes|k\rangle),$$
$$|e'_j\rangle := \sum_k |k\rangle\otimes\left(E'_j|k\rangle\right) = \sum_k (\mathrm{I}\otimes E'_j)(|k\rangle\otimes|k\rangle).$$

A straightforward computation using $\mathcal{E}=\mathcal{E}'$ shows that

$$\sum_i |e_i\rangle\langle e_i| = \sum_j |e'_j\rangle\langle e'_j|.$$

By Exercise 1.13, there exists a unitary matrix $U=(u_{ij})$ such that

$$|e_i\rangle = \sum_j u_{ij}|e'_j\rangle. \tag{1.5}$$

To show that $E_i = \sum_j u_{ij}E'_j$, it suffices to verify that the two sides agree on the basis vectors $\{|k\rangle\}$. Writing (1.5) explicitly, we have

$$\sum_k |k\rangle\otimes\left(E_i|k\rangle\right) = \sum_j u_{ij}\sum_k |k\rangle\otimes\left(E'_j|k\rangle\right).$$

Rearranging terms,

$$\sum_k |k\rangle\otimes\left(E_i|k\rangle\right) = \sum_k |k\rangle\otimes\Big(\sum_j u_{ij}E'_j|k\rangle\Big).$$

Since the vectors $\{|k\rangle\}$ are linearly independent, it follows that (see Exercise 1.13)

$$E_i |k\rangle = \sum_j u_{ij} E'_j |k\rangle \quad \text{for each } k.$$

Thus $E_i = \sum_j u_{ij} E'_j$.
Conversely, assume that $\{E_1, \ldots, E_r\}$ and $\{E'_1, \ldots, E'_r\}$ satisfy $E_i = \sum_j u_{ij} E'_j$ for some unitary matrix $U = (u_{ij})$. Then

$$\begin{aligned}
\sum_i E_i \rho E_i^\dagger &= \sum_i \Big(\sum_{j=1}^r u_{ij} E'_j \Big) \rho \Big(\sum_k \overline{u_{ik}} {E'_k}^\dagger \Big) \\
&= \sum_{i,j,k} \overline{u_{ik}} u_{ij} E'_j \rho {E'_k}^\dagger \\
&= \sum_{j,k} \delta_{jk} E'_j \rho {E'_k}^\dagger \\
&= \sum_j E'_j \rho {E'_j}^\dagger .
\end{aligned}$$

Hence, the two sets of Kraus operators define the same map. □

Notes

- Interested readers are encouraged to consult Chapter 2 and Section 8.2 of Nielsen and Chuang 2010 as well as Section 4 of Kribs 2005 for a more detailed treatment of the material covered in Lectures 1 and 2.

References for Lecture 1 & 2.

Kribs, David W. (2005). "A quantum computing primer for operator theorists". In: *Linear Algebra and its Applications* 400, pp. 147–167. ISSN: 0024-3795.

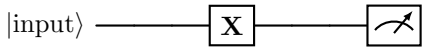
Nielsen, Michael A. and Isaac L. Chuang (2010). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press.

2. Lecture 3 & 4: Basics of quantum error correction and the QECC condition

Let us consider a simple example of using a quantum computer for computations. Suppose the input is either 0 or 1, and we want to compute the NOT of the input. The following quantum circuit[2] (see 5.2) accomplishes this, where $\mathbf{X}$ denotes the operation

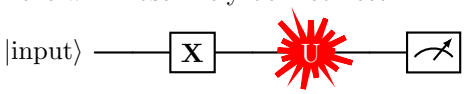
$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

and the meter symbol indicates measurement in the computational basis, i.e., $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$.



After performing the measurement, we collect statistics of the outcomes. From this statistical data, we infer the result.

Now consider the following situation. During the above experiment, an unknown quantum operation U is randomly applied to the circuit, which is unintended. Then, before the measurement, the state becomes $U\mathbf{X}|\text{input}\rangle$, i.e., it is "corrupted" by U. As a result, the inferred outcome from the measurement will most likely be incorrect.



The goal of quantum error correction is to recover the original data, provided we have some information or assumptions about the randomly applied operation U.

Exercise 2.1. ★ *Show that a quantum state $|\psi\rangle$ and $c|\psi\rangle$, for some scalar c, yield the same measurement statistics with respect to any measurement operators.*

Notation 2.2. If two quantum states $|\psi\rangle$ and $|\phi\rangle$ differ by a scalar multiple, we write $|\psi\rangle \propto |\phi\rangle$. The same notation applies to mixed states as well. In this case, we say that the two quantum states are the same up to a global phase. For all practical purposes, two quantum states can be considered equal if they differ only by a global phase.

Before we come to the quantum error correction, we will first try to understand how classical error correction works.

2.1. Classical Codes.

Let $\mathbb{Z}_2$ denote the finite cyclic group of order 2, and let $\mathbb{Z}_2^n$ denote the n-fold direct sum of this group. Classical computers use strings of *two-bit* information, i.e., elements of $\mathbb{Z}_2^n$.

The goal of communication is for a sender to transmit information through a channel to a receiver. However, the information may become *corrupted* during transmission due to *errors* in the channel. Error correction aims to recover the original data so that the receiver obtains the correct information.

By *encoding*, we mean a one-to-one map

$$i : \mathbb{Z}_2^k \hookrightarrow \mathbb{Z}_2^n, \quad \text{where } n \geq k.$$

To distinguish the two types of information, we refer to the domain $\mathbb{Z}_2^k$ as the *logical information*, and the codomain $\mathbb{Z}_2^n$ as the *physical information*. The image of i is called the *codespace*

[2]For these notes, a detailed understanding of quantum circuit diagrams is not required. Nevertheless, readers are encouraged to consult Kaye, Laflamme, and Mosca 2006, Chapter 4 for a more comprehensive introduction to quantum circuits.

(or just *code*), denoted by C. A *linear code* is a code where the encoding map is linear. Since the encoding map encodes k bits of logical data (information) into n bits of physical data (information), we say that the code C has parameters $[n, k]$.

As the term suggests, physical information interacts with the environment and is therefore susceptible to errors. Mathematically, an error E is a map from the physical information space $\mathbb{Z}_2^n$ to itself. In the absence of error, we take E to be the identity map I.

Let us describe this situation in a real-world context. Suppose the sender has a state (data) $x \in \mathbb{Z}_2^k$ that they wish to transmit to the receiver. To do so, they encode the data using the map i, obtaining the physical state $i(x)$. For simplicity, we denote this physical state by x again, when no confusion arises.

The encoded state is then transmitted through a "noisy channel" $\mathcal{E}$ to the receiver. Assume that the encoded state x undergoes an error E with probability p, or remains unchanged (i.e., is acted upon by the identity I) with probability $1 - p$. Hence, the receiver receives the state

$$\mathcal{E}(x) = \begin{cases} E(x), & \text{with probability } p, \\ x, & \text{with probability } 1 - p. \end{cases}$$

The receiver knows only the structure of the codespace and must attempt to recover the original logical information from the possibly corrupted physical state.

Before we present an example illustrating the above situation, let us first introduce some formal definitions.

Definition 2.3. *A (classical) channel $\mathcal{E} = \mathcal{E}(I, E_1, E_2, E_3, \ldots, E_r)$ is a collection of operators $E_i : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, for $i = 1, 2, \ldots, r$, together with the identity map I. Each operator E_i is referred to as an error.*

Definition 2.4. *An error E is detectable for a codespace $C \subseteq \mathbb{Z}_2^n$ if for $x \in C$, for $x, y \in C$, and $x \neq y$ imply $E(x) \neq y$.*

Exercise 2.5. ★ *An error E is detectable iff for $x \in C$, either $E(x) = x$ or $E(x) \notin C$.*

We now come to the definition of correctability of errors.

Definition 2.6. *A channel $\mathcal{E} = \mathcal{E}(I, E_1, E_2, E_3, \ldots, E_r)$ is said to be correctable for a code $C \subseteq \mathbb{Z}_2^n$ if for every y in the range of $\mathcal{E}$, there exists a unique $x \in C$ and some error operator E_i such that $E_i(x) = y$.*

Theorem 2.7. *The channel $\mathcal{E} = \mathcal{E}(I, E_1, E_2, E_3, \ldots, E_r)$ is correctable for a code C if and only if, for all $x, y \in C$ with $x \neq y$, we have $E_i(x) \neq E_j(y)$ for all i, j.*

Proof. Suppose $\mathcal{E}$ is correctable. Let $x, y \in C$ with $x \neq y$, and suppose for contradiction that $E_i(x) = E_j(y)$ for some i, j. Then there are two distinct codewords mapping to the same output under the channel, violating the uniqueness condition in the definition of correctability.

Conversely, assume that for all $x, y \in C$ with $x \neq y$, we have $E_i(x) \neq E_j(y)$ for all i, j. Let z be an output in the range of $\mathcal{E}$. Then there exists some $x \in C$ and some E_i such that $E_i(x) = z$. Suppose there is another $y \in C$ and E_j such that $E_j(y) = z$. By assumption, this implies $x = y$, ensuring uniqueness. Hence, $\mathcal{E}$ is correctable. □

Example 2.8. *[The repetition code]* This is the most basic example of an error-correcting code. In this example, the logical information is given by the two-element group $\mathbb{Z}_2$, with states denoted by 0 and 1. The information is encoded into the physical space $\mathbb{Z}_2^3$, whose

elements we denote by 3-bit binary strings such as 000, 010, 111, etc. We define the encoding map i by

$$i(0) = 000, \quad i(1) = 111.$$

Thus, the codespace is $C = \{000, 111\}$.

We now define the following error operators acting on $\mathbb{Z}_2^3$:

- One-bit-flip errors:
$$E_{11}(x, y, z) = (\mathrm{NOT}(x), y, z),$$
$$E_{12}(x, y, z) = (x, \mathrm{NOT}(y), z),$$
$$E_{13}(x, y, z) = (x, y, \mathrm{NOT}(z)).$$
- Two-bit-flip errors:
$$E_{21}(x, y, z) = (\mathrm{NOT}(x), \mathrm{NOT}(y), z), \qquad E_{22}(x, y, z) = (x, \mathrm{NOT}(y), \mathrm{NOT}(z)),$$
$$E_{23}(x, y, z) = (\mathrm{NOT}(x), y, \mathrm{NOT}(z)).$$
- Three-bit-flip error:
$$E_3(x, y, z) = (\mathrm{NOT}(x), \mathrm{NOT}(y), \mathrm{NOT}(z)).$$

Here, the NOT operation flips the bit: $\mathrm{NOT}(0) = 1$ and $\mathrm{NOT}(1) = 0$, and $x, y, z \in \mathbb{Z}_2$.

We now examine the action of these error operators on the codewords:

$$\begin{aligned}
E_{11}(000) &= (1, 0, 0), & E_{11}(111) &= (0, 1, 1),\\
E_{12}(000) &= (0, 1, 0), & E_{12}(111) &= (1, 0, 1),\\
E_{13}(000) &= (0, 0, 1), & E_{13}(111) &= (1, 1, 0),\\
E_{21}(000) &= (1, 1, 0), & E_{21}(111) &= (0, 0, 1),\\
E_{22}(000) &= (0, 1, 1), & E_{22}(111) &= (0, 1, 1),\\
E_{23}(000) &= (1, 0, 1), & E_2(111) &= (0, 1, 0),\\
E_3(000) &= (1, 1, 1), & E_3(111) &= (0, 0, 0).
\end{aligned}$$

From this, we observe:

- The one-bit-flip errors E_{11}, E_{12}, and E_{13} are *detectable*, since they map codewords to states outside the codespace.
- The two-bit-flip error E_{21}, E_{22}, and E_{23} are also detectable, for the same reason.
- The three-bit-flip error E_3 is *not detectable*, as it maps one codeword to another: $E_3(000) = 111$ and $E_3(111) = 000$.

As for correctability:

- The channel $\mathcal{E} = \mathcal{E}(\mathrm{id}, E_{11}, E_{12}, E_{13})$ is correctable, since each corrupted state has a unique pre-image in the codespace.
- The channel $\mathcal{E} = \mathcal{E}(\mathrm{id}, E_{11}, E_{12}, E_{13}, E_{21})$ is *not* correctable. For instance, the state $(0, 0, 1)$ can result from both $E_{13}(000)$ and $E_{21}(111)$, violating the uniqueness condition for correction.
- Any channel that includes E_3 is also not correctable, because E_3 is not even detectable.

Decoding strategy: Majority voting.

The decoding strategy we use here is called *majority decoding*. Given a received 3-bit string, the decoder outputs:

- 0 if two or more bits are 0

- 1 if two or more bits are 1

This strategy correctly decodes the original bit as long as at most one bit is flipped during transmission. If two or more bits are flipped, majority decoding fails.

Threshold computation. Suppose a single bit is flipped independently with probability p during transmission. The probabilities of different error types are:

$$P_0 = (1-p)^3, \quad \text{(no bit flip)}$$
$$P_1 = 3p(1-p)^2, \quad \text{(one bit flip)}$$
$$P_2 = 3p^2(1-p), \quad \text{(two bit flips)}$$
$$P_3 = p^3, \quad \text{(three bit flips)}.$$

The repetition code can correct all one-bit errors, but not two- or three-bit errors. Therefore, the code succeeds in correcting the message if 0 or 1 bits are flipped. So the success probability is:

$$P_{\text{success}} = P_0 + P_1 = (1-p)^3 + 3p(1-p)^2.$$

We say the code works better than doing nothing if

$$(1-p)^3 + 3p(1-p)^2 > 1-p.$$

Exercise 2.9. ★ *Show that the above inequality holds for $p < p_{th} = \frac{1}{2}$.*

Code distance. Suppose $x, y \in \mathbb{Z}_2^n$. Then the (Hamming) distance $d(x,y)$ between x and y is defined to be the number of positions x and y differ. One can easily check that this defines a metric on $\mathbb{Z}_2^n$.

We are now in a position to define the Hamming distance of a code C. Define the Hamming distance D_C by

$$D_C = D := \min_{x,y \in C, x \neq y} d(x,y).$$

A $[n,k]$ code with distance D we call an $[n,k,D]$ code.

Exercise 2.10. ★ *Show that for a linear code C, the distance $D = \min_{x \in C, x \neq 0} d(x,0)$. The number $d(x,0)$ is also called the weight of the element $x \in C$. The weight $d(x,0)$ is also simply denoted by* $\text{wt}(x)$.

For the following proposition we define the weight of an error. An error E with weight at most t, denoted by $\text{wt}(E) \leq t$, is an error which changes any element $x \in \mathbb{Z}_2^n$ at most t positions. That is, $d(x, E(x)) \leq t$.

Theorem 2.11. *A channel $\mathcal{E} = \mathcal{E}(I, E_1, E_2, E_3, \cdots)$ is correctable for a linear $[n,k,D]$ code C with $D \geq 2t+1$, for some integer t, if $\text{wt}(E_i) \leq t$, for all i.*

Proof. From the definition of correctability of errors, we have to show that if for all y in the range of $\mathcal{E}$, there exists a unique $x \in C$ such that $E_i(x) = y$ for some E_i. Suppose $x_1, x_2 \in C$ giving $E_i(x_1) = y$ and $E_j(x_2) = y$. Note that by hypothesis $d(x_i, y) \leq t$ for $i = 1, 2$. Then

$$d(x_1, x_2) \leq d(x_1, y) + d(x_2, y) \leq 2t < D.$$

This forces $x_1 = x_2$, which proves our claim. □

For detectability, we have the following result.

Exercise 2.12. ★ *Any error E with $\text{wt}(E) \leq 2t$ for a $[n,k,D]$ code C with $D \geq 2t+1$, for some integer t, is detectable.*

We have a following simple bound on the distance D of a code.

Exercise 2.13. ★ *If C is an $[n,k,D]$ code then $D \leq n-k+1$.*

Exercise 2.14. ★ *Let C be a code given by the kernel of the matrix*

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

Show that C is a $[7,4,3]$ code. This is also called Hamming code.

2.2. **Quantum codes.** Let $\mathcal{H}$ denote the finite-dimensional Hilbert space $\mathbb{C}^n$. An element of $\mathcal{B}(\mathcal{H}) = \mathrm{M}_n(\mathbb{C})$ will sometimes be referred to as an *error*.

A *quantum code* $\mathcal{C}$ is defined as a subspace of the Hilbert space $\mathcal{H}$. Let $P_{\mathcal{C}}$ denote the orthogonal projection operator that projects onto $\mathcal{C} \subseteq \mathcal{H}$.

Definition 2.15. *An error $E \in \mathrm{M}_n(\mathbb{C})$ is detectable for $\mathcal{C}$ if $\langle\phi|\psi\rangle = 0$ then $\langle\phi| E |\psi\rangle = 0$, for all $|\phi\rangle, |\psi\rangle \in \mathcal{C}$.*

Let us first prove a theorem which is a simple consequence of Definition 2.15.

Theorem 2.16. *Given an error $E \in \mathrm{M}_n(\mathbb{C})$, a constant $\lambda_E \in \mathbb{C}$, and a code $\mathcal{C}$, the following are equivalent.*

(1) The error E is detectable for $\mathcal{C}$, i.e. if $\langle\phi|\psi\rangle = 0$ then $\langle\phi| E |\psi\rangle = 0$, for all $|\phi\rangle, |\psi\rangle \in \mathcal{C}$.
(2) $P_{\mathcal{C}} E |\psi\rangle = \lambda_E |\psi\rangle$, for all $|\psi\rangle \in C$.
(3) $P_{\mathcal{C}} E P_{\mathcal{C}} = \lambda_E P_{\mathcal{C}}$.
(4) $\langle\phi| E |\psi\rangle = \lambda_E \langle\phi|\psi\rangle$, for all $|\phi\rangle, |\psi\rangle \in \mathcal{C}$.

Proof. • To show (2) implies (3), take any $|\psi\rangle$ in $\mathcal{H}$. Then we have,

$$\begin{aligned} P_{\mathcal{C}} E P_{\mathcal{C}} |\psi\rangle &= P_{\mathcal{C}} E (P_{\mathcal{C}} |\psi\rangle) \\ &= \lambda_E P_{\mathcal{C}} |\psi\rangle . \quad \text{(using (2))} \end{aligned}$$

Hence $P_{\mathcal{C}} E P_{\mathcal{C}} = \lambda_E P_{\mathcal{C}}$.

• Let us now show that (3) implies (4). For all $|\phi\rangle, |\psi\rangle \in \mathcal{C}$,

$$\begin{aligned} \langle\phi| E |\psi\rangle &= \langle\phi| P_{\mathcal{C}}^{\dagger} E |\psi\rangle \\ &= \langle\phi| P_{\mathcal{C}} E |\psi\rangle \\ &= \langle\phi| P_{\mathcal{C}} E P_{\mathcal{C}} |\psi\rangle \\ &= \langle\phi| \lambda_E P_{\mathcal{C}} |\psi\rangle \quad \text{(using (3))} \\ &= \lambda_E \langle\phi|\psi\rangle . \end{aligned}$$

Hence (4) is proved.

• (4) $\implies$ (1) is trivial.

• Now let us show that (1) implies (4). Choose an orthonormal basis $\{|\psi_i\rangle\}_i$ of $\mathcal{C}$. Now for $i \neq j$, We first show that $\langle\psi_i| E |\psi_i\rangle = \langle\psi_j| E |\psi_j\rangle$, for all i, j. To show this, let us define the following vectors:

$$|+_{ij}\rangle := |\psi_i\rangle + |\psi_j\rangle , \quad |-_{ij}\rangle := |\psi_i\rangle - |\psi_j\rangle .$$

(Here we omit the normalization factor.) Then it is easy to check that $\langle+_{ij}|-_{ij}\rangle = 0$, using the orthonormality of $|\psi_i\rangle$ and $|\psi_j\rangle$. Now using (4), this means that $\langle+_{ij}| E |-_{ij}\rangle = 0$. But if

we expand the expression $\langle +_{ij}| E |-_{ij}\rangle$, again using (4) we get that $\langle \psi_i| E |\psi_i\rangle = \langle \psi_j| E |\psi_j\rangle$. This means that the expression $\langle \psi_i| E |\psi_i\rangle$ is constant which does not depend on i. We can now define

$$\lambda_E := \langle \psi_i| E |\psi_i\rangle .$$

Now one can easily check (4) using this λ_E, and expanding $|\phi\rangle, |\psi\rangle$ in terms of the orthonormal basis $\{|\phi\rangle\}_i$ of $\mathcal{C}$.

• Finally we will show that (4) $\implies$ (2). Using (4) we know that for all $|\phi\rangle, |\psi\rangle \in \mathcal{C}$, $\langle \phi| E |\psi\rangle = \lambda_E \langle \phi|\psi\rangle$. Now

$$\begin{aligned} \langle \phi| E |\psi\rangle &= \langle \phi| P_{\mathcal{C}}^{\dagger} E |\psi\rangle \\ &= \langle \phi| P_{\mathcal{C}} E |\psi\rangle \end{aligned}$$

Hence $\langle \phi| P_{\mathcal{C}} E |\psi\rangle = \lambda_E \langle \phi|\psi\rangle = \langle \phi|\lambda_E \psi\rangle$. So we get $\langle \phi| (P_{\mathcal{C}} E - \lambda_E) |\psi\rangle = 0$, for all $|\phi\rangle, |\psi\rangle \in \mathcal{C}$. Fixing $|\psi\rangle$ and putting $|\phi\rangle = (P_{\mathcal{C}} E - \lambda_E) |\psi\rangle$ in the equation $\langle \phi| (P_{\mathcal{C}} E - \lambda_E) |\psi\rangle = 0$, we get $(P_{\mathcal{C}} E - \lambda_E) |\psi\rangle = 0$ which gives (2).

□

Exercise 2.17. ★ *Let U be a unitary operator acting on $\mathcal{H}$. Show that for a code $\mathcal{C} \subseteq \mathcal{H}$, E is detectable for $\mathcal{C}$ iff $UEU^{\dagger}$ is detectable for the code $U(\mathcal{C})$.*

Exercise 2.18. ★ *Show that the set of detectable errors for a code $\mathcal{C}$ form a linear subspace of $\mathrm{M}_n(\mathbb{C})$.*

Now we come to the concept of correctability of errors. Let us recall that a quantum channel $\mathcal{E}$ is given by a set of Kraus operators (errors) $\{E_i\}$.

Definition 2.19. *Let $\mathcal{C}$ be a code of a Hilbert space $\mathcal{H}$ and $\mathcal{E}$ be a quantum channel acting on $\mathcal{B}(\mathcal{H})$. Then we say $\mathcal{E}$ is correctable if there exists a trace preserving quantum channel $\mathcal{R}$ acting on $\mathcal{B}(\mathcal{H})$ such that*

$$\mathcal{R} \circ \mathcal{E}(\rho) \propto \rho, \quad \text{for all } \rho \text{ with } \rho = P_{\mathcal{C}} \rho P_{\mathcal{C}}.$$[3]

Theorem 2.20. *Let $\mathcal{C}$ be a code in a Hilbert space $\mathcal{H}$, and let $\mathcal{E}$ be a quantum channel acting on $\mathcal{B}(\mathcal{H})$. Assume that $\mathcal{E}$ is composed of noise operators $\{E_i\}$. Then $\mathcal{E}$ is correctable if and only if the operators $\{E_i^{\dagger} E_j\}_{i,j}$ are detectable.*

Proof. Let us assume that $\mathcal{E}$ is correctable. Then, by the definition of correctability, there exists a channel $\mathcal{R}$ such that

$$\mathcal{R} \circ \mathcal{E}(\rho) \propto \rho, \quad \text{for all } \rho \text{ with } \rho = P_{\mathcal{C}} \rho P_{\mathcal{C}}.$$

Now we define the operator $\mathcal{E}_{\mathcal{C}}$ acting on $\mathcal{B}(\mathcal{H})$ by

$$\mathcal{E}_{\mathcal{C}}(\rho) := \mathcal{E}(P_{\mathcal{C}} \rho P_{\mathcal{C}}), \quad \text{for } \rho \in \mathcal{B}(\mathcal{H}).$$

Let us first show that $\mathcal{E}_{\mathcal{C}}$ defines a completely positive map. If ρ is positive, then $P_{\mathcal{C}} \rho P_{\mathcal{C}}$ is also positive. Since $\mathcal{E}$ is positive, this implies that

$$\mathcal{E}_{\mathcal{C}}(\rho) = \mathcal{E}(P_{\mathcal{C}} \rho P_{\mathcal{C}})$$

is positive.

[3] By $\rho \propto \rho'$ we mean $\rho = c\rho'$, for some $c \in \mathbb{C}$.

Now consider a matrix $\rho = (\rho_{ij}) \in \mathrm{M}_l(\mathcal{B}(\mathcal{H}))$ that is positive. Then

$$\begin{aligned}\mathcal{E}_{\mathcal{C}}^{(l)}(\rho) &= (\mathcal{E}_{\mathcal{C}}(\rho_{ij})) \\ &= (\mathcal{E}(P_{\mathcal{C}}\rho_{ij}P_{\mathcal{C}})) \\ &= \mathcal{E}^{(l)}\left((P_{\mathcal{C}} \otimes \mathrm{I})\rho(P_{\mathcal{C}} \otimes \mathrm{I})\right),\end{aligned}$$

which is positive by the complete positivity of $\mathcal{E}$. This shows that the map $\mathcal{E}_{\mathcal{C}}$ is completely positive.

Next, note that $\mathrm{Tr}(\mathcal{E}_{\mathcal{C}}(\rho)) \leq \mathrm{Tr}(\rho)$ for any positive $\rho \in \mathcal{B}(\mathcal{H})$. Indeed, we have

$$\begin{aligned}\mathrm{Tr}(\mathcal{E}_{\mathcal{C}}(\rho)) &= \mathrm{Tr}(\mathcal{E}(P_{\mathcal{C}}\rho P_{\mathcal{C}})) \\ &\leq \mathrm{Tr}(P_{\mathcal{C}}\rho P_{\mathcal{C}}) \\ &= \mathrm{Tr}(P_{\mathcal{C}}\rho) \\ &\leq \mathrm{Tr}(\rho),\end{aligned}$$

since $\mathcal{E}$ is trace-preserving, and $P_{\mathcal{C}}$ is a projection.

(Recall that $\mathrm{Tr}(AB) \geq 0$ for positive operators A and B, since

$$\begin{aligned}\mathrm{Tr}(AB) &= \mathrm{Tr}(\sqrt{A}\sqrt{A}\sqrt{B}\sqrt{B}) \\ &= \mathrm{Tr}(\sqrt{B}\sqrt{A}\sqrt{A}\sqrt{B}) \\ &= \mathrm{Tr}\left((\sqrt{A}\sqrt{B})^{\dagger}\sqrt{A}\sqrt{B}\right) \geq 0.)\end{aligned}$$

Now, since $\mathcal{R} \circ \mathcal{E}(\rho) = \rho$ for all ρ satisfying $\rho = P_{\mathcal{C}}\rho P_{\mathcal{C}}$, we have

$$\mathcal{R} \circ \mathcal{E}_{\mathcal{C}}(\rho) = \mathcal{R} \circ \mathcal{E}(P_{\mathcal{C}}\rho P_{\mathcal{C}}) \propto P_{\mathcal{C}}\rho P_{\mathcal{C}}.$$

Expanding the above expression, we get

$$\sum_{i,j} R_j E_i P_{\mathcal{C}} \rho P_{\mathcal{C}} E_i^{\dagger} R_j^{\dagger} = \mu P_{\mathcal{C}}\rho P_{\mathcal{C}},$$

for some $\mu > 0$. This indicates that the collections $\{R_j E_i P_{\mathcal{C}}\}$ and $\{\sqrt{\mu}P_{\mathcal{C}}\}$ define the same completely positive map. Hence, using Proposition 1.19, there exist complex numbers α_{ki} such that

$$R_k E_i P_{\mathcal{C}} = \alpha_{ki}\sqrt{\mu}P_{\mathcal{C}}, \quad \text{for all } k, i.$$

Fix k. Then for indices i and j, we have

$$R_k E_i P_{\mathcal{C}} = \alpha_{ki}\sqrt{\mu}P_{\mathcal{C}}, \quad R_k E_j P_{\mathcal{C}} = \alpha_{kj}\sqrt{\mu}P_{\mathcal{C}}.$$

Hence,

$$P_{\mathcal{C}} E_i^{\dagger} R_k^{\dagger} R_k E_j P_{\mathcal{C}} = (\alpha_{ki})^* \alpha_{kj} \mu P_{\mathcal{C}}.$$

Summing over k, and using the relation $\sum_k R_k^{\dagger} R_k = \mathrm{I}$, we obtain

$$P_{\mathcal{C}} E_i^{\dagger} E_j P_{\mathcal{C}} = \mu \lambda_{ij} P_{\mathcal{C}},$$

where $\lambda_{ij} := \sum_k (\alpha_{ki})^* \alpha_{kj}$ is a constant. The above equation shows that $E_i^{\dagger}E_j$ is detectable for all i, j.

To prove the converse direction, assume that the operators $\{E_i^{\dagger}E_j\}_{i,j}$ are detectable. We aim to construct a trace-preserving channel $\mathcal{R}$ such that

$$\mathcal{R} \circ \mathcal{E}(\rho) \propto \rho, \quad \text{for all } \rho \text{ with } \rho = P_{\mathcal{C}}\rho P_{\mathcal{C}}.$$

Since the operators $\{E_i^\dagger E_j\}_{i,j}$ are detectable, there exist constants λ_{ij} such that

$$P_{\mathcal{C}} E_i^\dagger E_j P_{\mathcal{C}} = \lambda_{ij} P_{\mathcal{C}}.$$

The matrix (λ_{ij}) is Hermitian, hence diagonalizable. Therefore, there exists a unitary matrix U such that $U^{-1}(\lambda_{ij})U$ is diagonal. Using Proposition 1.19, and replacing $\mathcal{E}$ by $U\mathcal{E}$ if necessary, we may assume that (λ_{ij}) is diagonal.

Define

$$\mu := \sum_i \lambda_{ii} \leq 1,$$

which follows from taking the trace on both sides of

$$\sum_i P_{\mathcal{C}} E_i^\dagger E_i P_{\mathcal{C}} = \sum_i \lambda_{ii} P_{\mathcal{C}},$$

and using the fact that $\sum_i E_i^\dagger E_i \leq \mathrm{I}$.

Now, consider the polar decomposition of $E_i P_{\mathcal{C}}$. There exists a unitary operator U_i such that

$$E_i P_{\mathcal{C}} = U_i \sqrt{P_{\mathcal{C}} E_i^\dagger E_i P_{\mathcal{C}}} = U_i \sqrt{\lambda_{ii} P_{\mathcal{C}}} = U_i \sqrt{\lambda_{ii}} P_{\mathcal{C}}.$$

Define the projections $P_i := U_i P_{\mathcal{C}} U_i^\dagger$. We first verify that the P_i are mutually orthogonal. Indeed, for $i \neq j$:

$$\begin{aligned} P_i P_j = P_i^\dagger P_j &= U_i P_{\mathcal{C}} U_i^\dagger U_j P_{\mathcal{C}} U_j^\dagger \\ &= \frac{U_i P_{\mathcal{C}} E_i^\dagger E_j P_{\mathcal{C}} U_j^\dagger}{\sqrt{\lambda_{ii}}\sqrt{\lambda_{jj}}} = 0. \end{aligned}$$

If necessary, we can add the projection onto the orthogonal complement of the sum $\sum P_i$ (defining $U_i = \mathrm{I}$ in that case), so that $\sum P_i = \mathrm{I}$.

We now define the recovery channel $\mathcal{R}$ using noise operators $U_i^\dagger P_i$:

$$\mathcal{R}(\rho) = \sum_i U_i^\dagger P_i \rho P_i U_i, \quad \rho \in \mathcal{B}(\mathcal{H}).$$

To show that $\mathcal{R}$ is a valid quantum channel, note that

$$\sum_i P_i U_i U_i^\dagger P_i = \sum_i P_i = \mathrm{I}.$$

Now let ρ be such that $\rho = P_{\mathcal{C}} \rho P_{\mathcal{C}}$. We check that $\mathcal{R} \circ \mathcal{E}(\rho) = \mu\rho$. That is,

$$\mathcal{R} \circ \mathcal{E}(\rho) = \sum_{i,l} U_i^\dagger P_i E_l \rho E_l^\dagger P_i U_i = \sum_{i,l} \delta_{il} \lambda_{ii} \rho = \rho \sum_i \lambda_{ii} = \mu\rho, \tag{2.1}$$

which follows from the identity:

$$\begin{aligned} U_i^\dagger P_i E_l \sqrt{\rho} &= U_i^\dagger U_i P_{\mathcal{C}} U_i^\dagger E_l \sqrt{\rho} \\ &= P_{\mathcal{C}} U_i^\dagger E_l \sqrt{\rho} \\ &= \frac{P_{\mathcal{C}} E_i^\dagger E_l P_{\mathcal{C}} \sqrt{\rho}}{\sqrt{\lambda_{ii}}} \\ &= \frac{\lambda_{il} P_{\mathcal{C}} \sqrt{\rho}}{\sqrt{\lambda_{ii}}} = \delta_{il} \sqrt{\lambda_{ii}} \sqrt{\rho}. \end{aligned}$$

Note that we used the fact that $\sqrt{\rho} = P_{\mathcal{C}}\sqrt{\rho}P_{\mathcal{C}}$, which follows from taking the square root on both sides of $\rho = P_{\mathcal{C}}\rho P_{\mathcal{C}}$. Also, observe that $P_{\mathcal{C}}$ commutes with both $P_{\mathcal{C}}\rho$ and ρ.

□

Remark 2.21. If $\{E_i\}$ are the error/noise operators of a correctable quantum channel $\mathcal{E}$, then from the theorem above, we know that the operators $\{E_i^\dagger E_j\}_{i,j}$ are detectable. Since the operators $\{E_i^\dagger E_j\}_{i,j}$ are detectable, there exist constants λ_{ij} such that

$$P_{\mathcal{C}} E_i^\dagger E_j P_{\mathcal{C}} = \lambda_{ij} P_{\mathcal{C}}, \tag{2.2}$$

where $P_{\mathcal{C}}$ is the projector onto the code subspace $\mathcal{C}$.

The equation above (Equation 2.2) is known as the Knill–Laflamme condition. In view of Theorem 2.16, the Knill–Laflamme condition can be equivalently expressed as

$$\langle\phi| E_i^\dagger E_j |\psi\rangle = \lambda_{ij} \langle\phi|\psi\rangle, \tag{2.3}$$

for all $|\phi\rangle, |\psi\rangle \in \mathcal{C}$.

Remark 2.22. It is easy to see that in the case of encoding a single qubit (which means that the dimension of $\mathcal{C}$ is 2), the conditions of Theorem 2.16 are equivalent to the following:

$$\langle 0_L| E |0_L\rangle = \langle 1_L| E |1_L\rangle, \quad \langle 0_L| E |1_L\rangle = \langle 1_L| E |0_L\rangle = 0, \tag{2.4}$$

where $\{|0_L\rangle, |1_L\rangle\}$ is a basis of $\mathcal{C}$.

To see this, first note that condition (4) of Theorem 2.16 clearly implies Equation 2.4. On the other hand, let us show that condition (1) of Theorem 2.16 follows from Equation 2.4.

Assume $\langle\phi|\psi\rangle = 0$ for some $|\phi\rangle$ and $|\psi\rangle$ in $\mathcal{C}$. Writing $|\phi\rangle = c_1|0_L\rangle + c_2|1_L\rangle$ and $|\psi\rangle = c_1'|0_L\rangle + c_2'|1_L\rangle$, the condition $\langle\phi|\psi\rangle = 0$ gives us $(c_1')^* c_1 + (c_2')^* c_2 = 0$. Now, using the conditions in Equation 2.4 and the relation $(c_1')^* c_1 + (c_2')^* c_2 = 0$, we again obtain $\langle\phi| E |\psi\rangle = 0$, which is what we needed to show.

Exercise 2.23. ★ *Let U be a unitary operator acting on $\mathcal{H}$. Then show that for a code $\mathcal{C} \subseteq \mathcal{H}$, a channel $\mathcal{E}$ with Kraus operators $\{E_1, E_2, \cdots, E_n\}$ is correctable for $\mathcal{C}$ iff the channel $U\mathcal{E}U^\dagger$ with Kraus operators $\{UE_1U^\dagger, UE_2U^\dagger, \cdots, UE_nU^\dagger\}$ is correctable for $U(\mathcal{C})$.*

2.3. **Examples of quantum codes.** In this section we will restrict ourself to $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$, i.e. the n- qubit Hilbert space. As before, let $\{|0\rangle, |1\rangle\}$ denote the standard orthonormal basis for the Hilbert space $\mathbb{C}^2$. We define the following operators on $\mathbb{C}^2$.

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

These operators satisfy the relations

$$\mathbf{XZ} = -\mathbf{ZX} \quad \text{and} \quad \mathbf{X}^2 = \mathbf{Z}^2 = I,$$

and they are unitary. We also define $\mathbf{Y} := i\mathbf{XZ} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$.

Define an n-qubit Pauli operator on $\mathcal{H}$ to be of the form $i^t E_1 \otimes E_2 \otimes \cdots \otimes E_n$, where each E_i belongs to the set $\{\mathrm{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$ and $0 \leq t \leq 4$. When there is no confusion, we write $i^t E_1 \otimes E_2 \otimes \cdots \otimes E_n$ as $i^t E_1 E_2 \cdots E_n$. For example, we write the 2-qubit operator $\mathbf{X} \otimes \mathbf{Y}$ as $\mathbf{XY}$. Then the set of all n-qubit Pauli operators forms a group G_n. Note that the order of the group G_n is 4^{n+1}. If we mod out its center (phases), we obtain a group $\widetilde{G}_n$ of order 4^n.

Exercise 2.24. ★ *The elements of $\widetilde{G}_n$ form a basis of $\mathcal{B}(\mathcal{H})$.*

The following is a useful concept for a code $\mathcal{C}$ and determines how effective the code is. For an n-qubit Pauli operator E on $\mathcal{H}^{\otimes n}$ of the form $i^t E_1 \otimes E_2 \otimes \cdots \otimes E_n$, we define the weight of E (denoted $\mathrm{wt}(E)$) as the number of indices $i = 1, 2, \ldots, n$ for which E_i is non-identity.

With the above notion of weight for a Pauli operator, we can define the distance of a code $\mathcal{C} \subseteq \mathcal{H}^{\otimes n}$. We have the following definition:

$$\mathrm{dist}(\mathcal{C}) := \min \{k \mid E \in G_n \text{ is not detectable with } \mathrm{wt}(E) = k\}.$$

We have some easy observations regarding the concept of the distance of a code.

Exercise 2.25. ★ *Let $\mathcal{C}_2$ be a subspace of a code $\mathcal{C}_1 \subseteq \mathcal{H}^{\otimes n}$. Then* $\mathrm{dist}(\mathcal{C}_2) \geq \mathrm{dist}(\mathcal{C}_1)$.

Proposition 2.26. *A code $\mathcal{C} \subseteq \mathcal{H}$ with distance* $\mathrm{dist}(\mathcal{C}) = d = 2t+1$ *can correct any channel $\mathcal{E}$ with Kraus operators $\{E_i\} \subseteq G_n$ such that* $\mathrm{wt}(E_i) \leq t$ *for each i.*

Proof. First, note that for $E, F \in G_n$, we have $\mathrm{wt}(E^\dagger) = \mathrm{wt}(E)$ and $\mathrm{wt}(EF) \leq \mathrm{wt}(E) + \mathrm{wt}(F)$. Now take any channel $\mathcal{E}$ with Kraus operators $\{E_i\} \subseteq G_n$ such that $\mathrm{wt}(E_i) \leq t$ for each i. Then for all i, j, we have $\mathrm{wt}(E_i^\dagger E_j) \leq 2t < d$. This implies, from the definition of distance, that $E_i^\dagger E_j$ is detectable. Hence, using Theorem 2.20, we conclude that $\mathcal{C}$ can correct $\mathcal{E}$. □

A code with distance $d = 2t + 1$ which encodes k qubits into n qubits is called an $[[n, k, d]]$ code.

Example 2.27. *(The bit-flip code)* We consider $n = 3$ and $k = 1$, which means that one qubit is encoded into three qubits. Let us take the codespace $\mathcal{C}$, which is spanned by the vectors $|000\rangle$ and $|111\rangle$. We claim that this code can correct the channel $\mathcal{E}$ consisting of the operators I, $E_1 = \mathbf{X}\mathrm{II}$, $E_2 = \mathrm{I}\mathbf{X}\mathrm{I}$, and $E_3 = \mathrm{II}\mathbf{X}$, i.e., a bit-flip on any one qubit. This can be readily checked using Equation 2.3.

Another way to state this is that the errors $\mathbf{XX}\mathrm{I}$, $\mathrm{I}\mathbf{XX}$, and $\mathbf{X}\mathrm{I}\mathbf{X}$ are detectable. For example, if we take $\phi = c_0 |000\rangle + c_1 |111\rangle$ and $\psi = c_0' |000\rangle + c_1' |111\rangle$, then

$$\langle \phi | \, I\mathbf{XX} \, | \psi \rangle = 0.$$

However, this code cannot detect any $\mathbf{Z}$ error on any qubit. For example, if we denote $\mathbf{Z}II$ by E, then on the one hand we have

$$\langle 000 | \, E \, | 000 \rangle = \langle 000 | 000 \rangle = 1,$$

giving $\lambda_E = 1$ (see Theorem 2.16), and on the other hand we have

$$\langle 111 | \, E \, | 111 \rangle = - \langle 111 | 111 \rangle = -1,$$

giving $\lambda_E = -1$. This is a contradiction.

Since the weight of the error $\mathbf{Z}\mathrm{II}$ is 1, the distance of the code $\mathcal{C}$ is 1. Hence, the code $\mathcal{C}$ is a $[[3, 1, 1]]$ code.

Example 2.28. *(The phase-flip code)* As in the previous example, let us take $n = 3$ and $k = 1$. We want to find a codespace that corrects $\mathbf{Z}$ errors. We have seen that the bit-flip code cannot correct $\mathbf{Z}$ errors.

Define the Hadamard transformation $\mathbf{H}$, which is unitary, on a single qubit by $\mathbf{H} |0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} =: |+\rangle$ and $\mathbf{H} |1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} =: |-\rangle$. Note that $\mathbf{HXH}^\dagger = \mathbf{Z}$, $\mathbf{HZH}^\dagger = \mathbf{X}$, and $\mathbf{H}^\dagger = \mathbf{H}$.

Then we have:

$$\mathbf{H}^{\otimes 3}(\mathbf{X}\mathrm{II})\mathbf{H}^{\otimes 3} = \mathbf{Z}\mathrm{II},$$
$$\mathbf{H}^{\otimes 3}(\mathrm{I}\mathbf{X}\mathrm{I})\mathbf{H}^{\otimes 3} = \mathrm{I}\mathbf{Z}\mathrm{I},$$
$$\mathbf{H}^{\otimes 3}(\mathrm{II}\mathbf{X})\mathbf{H}^{\otimes 3} = \mathrm{II}\mathbf{Z}.$$

Hence, in view of Corollary 2.23, the code $\mathbf{H}^{\otimes 3}(\mathcal{C})$ corrects a channel with Kraus operators $\mathbf{Z}II$, $I\mathbf{Z}I$, and $II\mathbf{Z}$, where $\mathcal{C}$ is the bit-flip code. Since $\mathbf{H}|0\rangle = |+\rangle$ and $\mathbf{H}|1\rangle = |-\rangle$, the code $\mathbf{H}^{\otimes 3}(\mathcal{C})$ is spanned by the vectors $|+++\rangle$ and $|---\rangle$.

As in the previous example, this code cannot detect any $\mathbf{X}$ error on any qubit. Hence, the phase-flip code $\mathbf{H}^{\otimes 3}(\mathcal{C})$ is a $[[3,1,1]]$ code.

Remark 2.29. We can also define an $n = 2$ and $k = 1$ version of the above two codes. In these cases, the codespaces $\mathcal{C}_1$ and $\mathcal{C}_2$ are spanned by the vectors $|00\rangle, |11\rangle$ and $|{+}{+}\rangle, |{-}{-}\rangle$, respectively. These codes are not very useful in the sense that they do not correct any error, although $\mathcal{C}_1$ does detect the error $\mathbf{X}\mathrm{I}$ or $\mathrm{I}\mathbf{X}$, and $\mathcal{C}_2$ does detect the error $\mathbf{Z}\mathrm{I}$ or $\mathrm{I}\mathbf{Z}$.

Example 2.30. *(The $[[4,1,2]]$ code)* This code $\mathcal{C}$ is generated by

$$|0_L\rangle = \frac{|0000\rangle + |1111\rangle}{\sqrt{2}}, \quad |1_L\rangle = \frac{|1100\rangle + |0011\rangle}{\sqrt{2}}.$$

The code $\mathcal{C}$ detects any single-qubit bit-flip error, i.e., the operators $\mathbf{X}\mathrm{III}$, $\mathrm{I}\mathbf{X}\mathrm{II}$, $\mathrm{II}\mathbf{X}\mathrm{I}$, and $\mathrm{III}\mathbf{X}$. This can be easily verified using the conditions of Theorem 2.16 (similar to the computations in Example 2.27).

To show that the code $\mathcal{C}$ detects any single-qubit phase-flip error, i.e., the operators $\mathbf{Z}\mathrm{III}$, $\mathrm{I}\mathbf{Z}\mathrm{II}$, $\mathrm{II}\mathbf{Z}\mathrm{I}$, and $\mathrm{III}\mathbf{Z}$, we can invoke Equation 2.4 from Remark 2.22. Let us consider only the case $E = \mathbf{Z}\mathrm{III}$. We compute the following quantities:

$$\langle 0_L|E|0_L\rangle, \quad \langle 1_L|E|1_L\rangle, \quad \langle 0_L|E|1_L\rangle, \quad \langle 1_L|E|0_L\rangle.$$

A simple computation shows that

$$\langle 0_L|E|0_L\rangle = \langle 1_L|E|1_L\rangle = 0, \quad \langle 0_L|E|1_L\rangle = \langle 1_L|E|0_L\rangle = 0.$$

Hence, Equation 2.4 shows that E is detectable by $\mathcal{C}$. The other cases $\mathrm{I}\mathbf{Z}\mathrm{II}$, $\mathrm{II}\mathbf{Z}\mathrm{I}$, and $\mathrm{III}\mathbf{Z}$ are similarly easy to check.

The code $\mathcal{C}$ also detects any single-qubit $\mathbf{Y}$ error, i.e., the operators $\mathbf{Y}\mathrm{III}$, $\mathrm{I}\mathbf{Y}\mathrm{II}$, $\mathrm{II}\mathbf{Y}\mathrm{I}$, and $\mathrm{III}\mathbf{Y}$, where $\mathbf{Y}$ is the product $\mathbf{X}\mathbf{Z}$ (note that we omit the scalar i in the definition of $\mathbf{Y}$). To show this, consider only the case $E = \mathbf{Y}\mathrm{III}$. Again, we compute

$$\langle 0_L|E|0_L\rangle, \quad \langle 1_L|E|1_L\rangle, \quad \langle 0_L|E|1_L\rangle, \quad \langle 1_L|E|0_L\rangle,$$

all of which evaluate to zero. Hence, $\mathbf{Y}\mathrm{III}$ is also detectable.

The above observations show that $\mathcal{C}$ can detect any single-qubit error. Now, observe that the code $\mathcal{C}$ cannot detect a two-qubit error. For example, if we take $E = \mathbf{X}\mathbf{X}\mathrm{II}$, then

$$\langle 0_L|\mathbf{X}\mathbf{X}\mathrm{II}|1_L\rangle \neq 0.$$

Hence, we conclude that $\mathcal{C}$ is a $[[4,1,2]]$ quantum code.

Example 2.31. *(Shor code)* This is the first quantum code, discovered by Shor, that can correct any single-qubit error. Let us go through the details.

The Shor code is given by

$$|0_L\rangle := \frac{|000000000\rangle + |111111000\rangle + |000111111\rangle + |111000111\rangle}{2\sqrt{2}},$$

$$|1_L\rangle := \frac{|111000000\rangle + |000000111\rangle + |000111000\rangle + |111111111\rangle}{2\sqrt{2}}.$$

One could also use the following orthonormal basis:

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

The error-detecting properties of the Shor code are very similar to those of the $[[4, 1, 2]]$ code.

Exercise 2.32. ★ *Show that the Shor code is a $[[9, 1, 3]]$ quantum code.*

2.4. **Examples of quantum channels.**

Example 2.33. *A one-qubit depolarizing quantum channel is defined as:*

$$\mathcal{E}(\rho) = \pi(\mathbf{X})\,\mathbf{X}\rho\mathbf{X} + \pi(\mathbf{Y})\,\mathbf{Y}\rho\mathbf{Y} + \pi(\mathbf{Z})\,\mathbf{Z}\rho\mathbf{Z} + (1 - \pi(\mathbf{X}) - \pi(\mathbf{Y}) - \pi(\mathbf{Z}))\,\rho,$$

where the values $\pi(\mathbf{X}), \pi(\mathbf{Y}), \pi(\mathbf{Z})$, and their sum $\pi(\mathbf{X}) + \pi(\mathbf{Y}) + \pi(\mathbf{Z})$ all lie in the interval $[0, 1]$.

The Kraus operators for this channel are:

$$E_0 = \mathrm{I}, \quad E_1 = \sqrt{\pi(\mathbf{X})}\,\mathbf{X}, \quad E_2 = \sqrt{\pi(\mathbf{Y})}\,\mathbf{Y}, \quad E_3 = \sqrt{\pi(\mathbf{Z})}\,\mathbf{Z}.$$

Physical interpretation: *When a quantum state ρ passes through the channel $\mathcal{E}$, it undergoes:*

- *an $\mathbf{X}$ (bit-flip) error with probability $\pi(\mathbf{X})$,*
- *a $\mathbf{Y}$ error with probability $\pi(\mathbf{Y})$,*
- *a $\mathbf{Z}$ (phase-flip) error with probability $\pi(\mathbf{Z})$,*
- *and remains unchanged with probability $1 - \pi(\mathbf{X}) - \pi(\mathbf{Y}) - \pi(\mathbf{Z})$.*

Special cases:

- *If $\pi(\mathbf{Y}) = \pi(\mathbf{Z}) = 0$, the channel is called a bit-flip channel.*
- *If $\pi(\mathbf{X}) = \pi(\mathbf{Y}) = 0$, the channel is called a phase-flip channel.*

A quantum channel $\mathcal{E}$ acting on n qubits is called an *independent channel* if it decomposes as: $\mathcal{E} = \mathcal{E}_1 \otimes \mathcal{E}_2 \otimes \cdots \otimes \mathcal{E}_n$, where each $\mathcal{E}_i$ is a single qubit channel.

In general, a *Pauli channel* on n qubits is given by:

$$\mathcal{E}_\pi(\rho) = \sum_{E \in G_n} \pi(E)\, E\rho E^\dagger,$$

where G_n is the n-qubit Pauli group, and $\pi : G_n \to [0, 1]$ is a probability distribution over G_n.

Example 2.34. *The amplitude damping channel, which models energy dissipation, is defined as:*

$$\mathcal{E}(\rho) = E_0\rho E_0^\dagger + E_1\rho E_1^\dagger,$$

with Kraus operators:

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad E_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix},$$

where $\gamma \in [0, 1]$.

Exercise 2.35. ★ *Write down explicit examples of quantum channels that are correctable by the bit-flip code, the $[[4, 1, 2]]$ code, and the Shor code.*

Notes

- The basic ideas of error correction for classical and quantum channels, as described above, can be found in Knill et al. 2002.
- The discussion of quantum codes, error detectability, and the Knill–Laflamme conditions is based on Kribs 2005.
- The examples of quantum codes and error-correction procedures presented above can be found in Chapter 10 of Nielsen and Chuang 2010.

References for Lecture 3 & 4.

Kaye, Phillip, Raymond Laflamme, and Michele Mosca (2006). *An Introduction to Quantum Computing*. Oxford, England: Oxford University Press UK.

Knill, E. et al. (2002). *Introduction to Quantum Error Correction*. eprint: arXiv:quant-ph/0207170.

Kribs, David W. (2005). "A quantum computing primer for operator theorists". In: *Linear Algebra and its Applications* 400, pp. 147–167. ISSN: 0024-3795.

Nielsen, Michael A. and Isaac L. Chuang (2010). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press.

3. Lecture 5 & 6: Stabilizer codes and CSS codes

We cannot clone, perforce; instead, we split
Coherence to protect it from that wrong
That would destroy our valued quantum bit
And make our computation take too long.

Correct a flip and phase - that will suffice.
If in our code another error's bred,
We simply measure it, then God plays dice,
Collapsing it to X or Y or zed.

We start with noisy seven, nine, or five
And end with perfect one. To better spot
Those flaws we must avoid, we first must strive
To find which ones commute and which do not.

With group and eigenstate, we've learned to fix
Your quantum errors with our quantum tricks.

- 'Quantum Error Correction Sonnet', by Daniel Gottesman

In this section, we introduce a class of quantum error-correcting codes that covers many important examples. These codes are known as *stabilizer codes*. The codes are defined as the subspace of the Hilbert space stabilized by a collection of Pauli operators.

Recall that the set of all n-qubit Pauli operators forms a group G_n. Let G be a subgroup of G_n. Then the stabilizer code $\mathcal{C}(G)$ is defined as

$$\mathcal{C}(G) := \{|\psi\rangle \in \mathcal{H} \mid g\,|\psi\rangle = |\psi\rangle\,,\ \forall g \in G\}.$$

Here $\mathcal{H}$ is the n-qubit Hilbert space. Let us quickly check that $\mathcal{C}(G)$ is a subspace of $\mathcal{H}$. If $|\psi_1\rangle$ and $|\psi_2\rangle$ are stabilized by some $g \in G$ (meaning $g\,|\psi\rangle = |\psi\rangle$), then any linear combination of $|\psi_1\rangle$ and $|\psi_2\rangle$ is also stabilized by g. Hence, $\mathcal{C}(G)$ is indeed a subspace of $\mathcal{H}$.

To ensure that $\mathcal{C}(G)$ is nontrivial, we impose two conditions: 1. $-\mathrm{I} \notin G$ (which also implies $\pm i\mathrm{I} \notin G$). 2. All elements of G commute with each other.

The first point is clear. For the second, suppose $g, h \in G$ anticommute, i.e., $gh = -hg$[4]. Then for every $|\psi\rangle \in \mathcal{C}(G)$ we have

$$gh\,|\psi\rangle = -hg\,|\psi\rangle\,.$$

But since $g\,|\psi\rangle = |\psi\rangle$ and $h\,|\psi\rangle = |\psi\rangle$, this implies $|\psi\rangle = 0$. Thus, for $\mathcal{C}(G)$ to be nontrivial, G must satisfy the above two conditions.

A subgroup G of G_n with these properties is called a *stabilizer group*.

It is now natural to ask: what is the dimension of $\mathcal{C}(G)$ for a given stabilizer group G? Since G is a finite group, we can always find $g_1, \ldots, g_l \in G$ such that

$$G = \langle g_1, g_2, \ldots, g_l\rangle,$$

meaning G is generated by the elements $g_1, \ldots, g_l$. We may further assume (using a simple induction argument) that $g_1, \ldots, g_l$ are *independent* in the following sense: if we remove any g_i from the list $g_1, \ldots, g_l$, then

$$\langle g_1, \ldots, g_{i-1}, g_{i+1}, \ldots, g_l\rangle \neq \langle g_1, \ldots, g_l\rangle.$$

[4]It is easy to check that g and h either commute or anticommute.

For such a set of generators, we can count the number of elements in G. Any element $g \in G$ can be written as

$$g = g_1^{a_1} g_2^{a_2} \cdots g_l^{a_l},$$

where $a_i \in \{0, 1\}$ for each i. This follows from the facts that all g_i commute with each other and that $g_i^2 = \mathrm{I}$ (since $-\mathrm{I} \notin G$).

Moreover, this representation of g is unique. To see this, suppose

$$g = g_1^{b_1} g_2^{b_2} \cdots g_l^{b_l}, \quad b_i \in \{0, 1\}.$$

If, without loss of generality, $a_1 \neq b_1$, then by commutativity we could express g_1 as a product of powers of $g_2, g_3, \ldots, g_l$. This contradicts the independence of $g_1, \ldots, g_l$.

Therefore, every element $g \in G$ has a unique representation of the above form, and hence G contains exactly 2^l elements.

The following theorem determines the dimension of $\mathcal{C}(G)$ for a given stabilizer group G.

Theorem 3.1. *Let G be a stabilizer subgroup of G_n generated by l independent elements $g_1, g_2, \ldots, g_l$, i.e., $G = \langle g_1, \ldots, g_l \rangle$. Then $\mathcal{C}(G)$ is a 2^{n-l}-dimensional vector space.*

Proof. We first claim that the orthogonal projection onto the subspace $\mathcal{C}(G)$, denoted by $P_{\mathcal{C}(G)}$, is given by

$$P := P_{\mathcal{C}(G)} = \frac{1}{2^l} \sum_{g \in G} g.$$

If $|\psi\rangle \in \mathcal{C}(G)$, then $g|\psi\rangle = |\psi\rangle$ for all $g \in G$, which in turn gives $P|\psi\rangle = |\psi\rangle$. Moreover, for any $g' \in G$,

$$g' P |\psi\rangle = \frac{1}{2^l} \sum_{g \in G} g' g |\psi\rangle = \frac{1}{2^l} \sum_{g \in G} (g' g) |\psi\rangle = P |\psi\rangle,$$

which shows that $P|\psi\rangle \in \mathcal{C}(G)$. Altogether, we have $P^2 = P$, and the range of P is precisely $\mathcal{C}(G)$. From the definition of P we also see that $P^\dagger = P$. This proves our claim.

Now, the dimension of the subspace $\mathcal{C}(G)$ is given by

$$\mathrm{Tr}(P) = \frac{1}{2^l} \sum_{g \in G} \mathrm{Tr}(g).$$

If $g = I$ then $\mathrm{Tr}(g) = 2^n$. If $g \neq \mathrm{I}$, then at least one tensor component of g must contain an $\mathbf{X}$, $\mathbf{Y}$, or $\mathbf{Z}$, and since $\mathrm{Tr}(\mathbf{X}) = \mathrm{Tr}(\mathbf{Y}) = \mathrm{Tr}(\mathbf{Z}) = 0$, it follows that $\mathrm{Tr}(g) = 0$. Thus,

$$\mathrm{Tr}(P) = \frac{1}{2^l} \sum_{g \in G} \mathrm{Tr}(g) = \frac{2^n}{2^l} = 2^{n-l},$$

which completes the proof. □

Example 3.2. Let us consider a simple example of a stabilizer code. Take the subgroup G inside G_3 generated by **ZZ**I and I**ZZ**. These clearly commute. In this case, the subspace $\mathcal{C}(G)$ must be $2^1 = 2$ dimensional. Consider the two independent vectors $|000\rangle$ and $|111\rangle$, which are both stabilized by **ZZ**I and I**ZZ**. Hence, by Theorem 3.1, $\mathcal{C}(G)$ is the subspace spanned by $|000\rangle$ and $|111\rangle$. This is exactly the codespace of the bit-flip code (Example 2.27).

3.1. **Check matrix for a stabilizer code.** Recall that an element g of the group G_n can be written as

$$g = c\,\mathbf{X}^{a_1}\mathbf{Z}^{b_1} \otimes \mathbf{X}^{a_2}\mathbf{Z}^{b_2} \otimes \cdots \otimes \mathbf{X}^{a_n}\mathbf{Z}^{b_n}, \tag{3.1}$$

where $a_g := (a_1, a_2, \ldots, a_n) \in \mathbb{Z}_2^n$, $b_g := (b_1, b_2, \ldots, b_n) \in \mathbb{Z}_2^n$, and $c \in \{\pm 1, \pm i\}$. Hence, up to an overall phase c, any element $g \in G_n$ can be represented as the vector $(a_g, b_g) \in \mathbb{Z}_2^{2n}$.

We define a map Ω from G_n to $\mathbb{Z}_2^{2n}$ by

$$\Omega(g) = (a_g, b_g) =: (a_g \mid b_g).$$

Clearly, this map ignores the phase of the Pauli operator g.

Exercise 3.3. ★ *Show that for two operators $g, h \in G_n$,*

$$\Omega(gh) = \Omega(g) + \Omega(h), \tag{3.2}$$

where $+$ denotes binary addition.

Exercise 3.4. ★ *Show that two operators $g, h \in G_n$ commute (resp. anticommute) if and only if*

$$\Omega(g) \begin{pmatrix} 0 & \mathrm{I} \\ \mathrm{I} & 0 \end{pmatrix} \Omega(h)^T = 0 \text{ (resp. } 1). \tag{3.3}$$

Suppose G is a stabilizer subgroup of G_n generated by the elements $g_1, g_2, \ldots, g_l$. We define the matrix $\Omega(G)$, called the *check matrix* of the subgroup G, as follows: $\Omega(G)$ is an $l \times 2n$ matrix whose i-th row is given by $\Omega(g_i)$.

For example, in the case of Example 3.2, where G is generated by $\mathbf{ZZ}I$ and $I\mathbf{ZZ}$, the corresponding check matrix $\Omega(G)$ is

$$\Omega(G) = \left(\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right).$$

Denote by J the $2n \times 2n$ (binary) symplectic form

$$J = \begin{pmatrix} 0 & \mathrm{I} \\ \mathrm{I} & 0 \end{pmatrix}.$$

(Since we are working over the binary field, we do not need the minus sign, which appears in the definition of a general symplectic form, in the symplectic form.) In view of Exercise 3.4, we clearly have

$$\Omega(g_i) J \Omega(g_k)^T = 0, \quad \forall\, g_i, g_k \in G.$$

We then have

$$\Omega(G) J \Omega(G)^T = 0. \tag{3.4}$$

Exercise 3.5. ★ *Let $G = \langle g_1, \ldots, g_l \rangle$ be a stabilizer group. Show that the generators $(g_i)_i$ are independent if and only if the rows of $\Omega(G)$ are linearly independent.*

Example 3.6. *(Steane code)* Consider the elements

$$g_1 = \mathrm{III}\mathbf{XXXX}, \quad g_2 = \mathrm{I}\mathbf{XX}\mathrm{II}\mathbf{XX}, \quad g_3 = \mathbf{X}\mathrm{I}\mathbf{X}\mathrm{I}\mathbf{X}\mathrm{I}\mathbf{X},$$

$$g_4 = \mathrm{III}\mathbf{ZZZZ}, \quad g_5 = \mathrm{I}\mathbf{ZZ}\mathrm{II}\mathbf{ZZ}, \quad g_6 = \mathbf{Z}\mathrm{I}\mathbf{Z}\mathrm{I}\mathbf{Z}\mathrm{I}\mathbf{Z}.$$

in G_7. Let G be the subgroup generated by these elements. The corresponding check matrix $\Omega(G)$ is

$$\Omega(G) = \left(\begin{array}{ccccccc|ccccccc} 0&0&0&1&1&1&1&0&0&0&0&0&0&0\\ 0&1&1&0&0&1&1&0&0&0&0&0&0&0\\ 1&0&1&0&1&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1\\ 0&0&0&0&0&0&0&0&1&1&0&0&1&1\\ 0&0&0&0&0&0&0&1&0&1&0&1&0&1 \end{array}\right).$$

It is straightforward to verify that the rows of $\Omega(G)$ are linearly independent and that all the generators g_i commute, using Equation 3.3. Therefore, by Theorem 3.1, the corresponding stabilizer code has parameters $[[7, 1]]$.

Example 3.7. *(Shor code)* The Shor code of Example 2.31 can also be described as a stabilizer code. For this, consider the following generators in G_9:

$$g_1 = \mathbf{ZZ}\text{IIIIIII}, \quad g_2 = \text{I}\mathbf{ZZ}\text{IIIIII}, \quad g_3 = \text{III}\mathbf{ZZ}\text{IIII},$$
$$g_4 = \text{IIII}\mathbf{ZZ}\text{III}, \quad g_5 = \text{IIIII}\mathbf{ZZ}\text{I}, \quad g_6 = \text{IIIIII}\mathbf{ZZ},$$
$$g_7 = \mathbf{XXXXXX}\text{III}, \quad g_8 = \text{III}\mathbf{XXXXXX}.$$

The check matrix $\Omega(G)$ of the group G generated by these elements is

$$\Omega(G) = \left(\begin{array}{ccccccccc|ccccccccc} 0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1\\ 1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0 \end{array}\right).$$

It is clear that the rows of $\Omega(G)$ are linearly independent. Moreover, all the generators commute. One can check that the stabilizer code $\mathcal{C}(G)$ defined by G is precisely the code described in Example 2.31.

3.2. **Error detection for the stabilizer codes.** Recall that the centralizer of a subgroup $G \subseteq G_n$, denoted $N(G)$, is defined as the set of all elements $E \in G_n$ such that $Eg = gE$ for all $g \in G$. Note that $N(G)$ is exactly the normalizer of G provided $-\text{I} \notin G$. Indeed, if $h \in N(G)$, then $hgh^{-1} \in G$. If instead $hg = -gh$, it would follow that $-\text{I} \in G$. Hence we must have $hg = gh$ for all $g \in G$.

Now, let G be a stabilizer subgroup of G_n. In that case, G is clearly a subset of $N(G)$. Furthermore, if we define

$$\text{c}G := \{\, kg \;\mid\; g \in G,\; k \in \mathbb{C} \,\},$$

then cG is also a subset of $N(G)$.

We have the following theorem.

Theorem 3.8. *Let G be a stabilizer subgroup of G_n. Then $E \in G_n$ is detectable for the code $\mathcal{C}(G)$ if and only if*

$$E \notin N(G) \setminus \text{c}G.$$

Proof. We first prove the "if" direction using condition (1) of Theorem 2.16.

- Suppose $E \in \mathrm{c}G$. Let $|\phi\rangle, |\psi\rangle \in \mathcal{C}(G)$ with $\langle\phi|\psi\rangle = 0$. Since E stabilizes $|\psi\rangle$, we have $\langle\phi| E |\psi\rangle = 0$.

- Now suppose $E \notin N(G)$. Since elements of the Pauli group either commute or anticommute, E must anticommute with some $g \in G$. Again let $|\phi\rangle, |\psi\rangle \in \mathcal{C}(G)$ with $\langle\phi|\psi\rangle = 0$. Then

$$\langle\phi| E |\psi\rangle = \langle\phi| Eg |\psi\rangle = \langle\phi| (-gE) |\psi\rangle = -\langle\phi| gE |\psi\rangle = -\langle\phi| E |\psi\rangle . \tag{3.5}$$

This implies $\langle\phi| E |\psi\rangle = 0$, showing that E is detectable by condition (1) of Theorem 2.16.

Now we prove the "only if" direction. Assume E is detectable and that $E \in N(G) \setminus \mathrm{c}G$. If $|\psi\rangle \in \mathcal{C}(G)$, then for all $g \in G$,

$$gE |\psi\rangle = Eg |\psi\rangle = E |\psi\rangle .$$

Hence $E |\psi\rangle \in \mathcal{C}(G)$, so $\mathcal{C}(G)$ is invariant under E.

By condition (2) of Theorem 2.16, we must have

$$E |\psi\rangle = \lambda_E |\psi\rangle ,$$

for some $\lambda_E \in \mathbb{C}$ independent of $|\psi\rangle$. Since E is unitary, $\lambda_E \neq 0$, and we may define $E' := \lambda_E^{-1} E$. Then $E' |\psi\rangle = |\psi\rangle$ for all $|\psi\rangle \in \mathcal{C}(G)$. Moreover, because $E^2 = \pm \mathrm{I}$, it follows that $\lambda_E \in \{\pm 1, \pm i\}$, hence $E' \in G_n$.

Note that $E' \notin G$, otherwise $E \in \mathrm{c}G$, which contradicts the assumption. Consider the subgroup generated by G and E'. This defines a stabilizer code strictly smaller than $\mathcal{C}(G)$. Therefore, there must exist some $|\psi\rangle \in \mathcal{C}(G)$ such that

$$E' |\psi\rangle \neq |\psi\rangle ,$$

which contradicts the fact that $E' |\psi\rangle = |\psi\rangle$ for all $|\psi\rangle \in \mathcal{C}(G)$.

Thus $E \notin N(G) \setminus \mathrm{c}G$, completing the proof. □

Remark 3.9. *Theorem 3.8 implies that we can define the distance of the code $\mathcal{C}(G)$ as*

$$\mathrm{Dist}(\mathcal{C}(G)) := \min\{\mathrm{wt}(E) \;\mid\; E \in N(G) \setminus \mathrm{c}G\}.$$

For any $u \in \mathbb{Z}_2^{2n}$, define the *symplectic weight* of u by

$$\mathrm{wt}_s(u) := \#\{\, j \in [n] \;\mid\; (u_j, u_{j+n}) \neq (0,0) \,\}.$$

It is easy to see that for any $g \in G_n$,

$$\mathrm{wt}(g) = \mathrm{wt}_s(\Omega(g)).$$

Using Remark 3.9, we obtain

$$\mathrm{Dist}(\mathcal{C}(G)) = \min\{\mathrm{wt}_s(v) \;\mid\; v \in \Omega(N(G)) \setminus \Omega(G)\}. \tag{3.6}$$

Example 3.10. We now give an example of a non-stabilizer code. Consider the code spanned by the vectors

$$\frac{1}{\sqrt{2}}\big(|01\rangle + |10\rangle\big), \quad |11\rangle .$$

This code cannot be a stabilizer code. The only non-trivial elements that stabilize $|11\rangle$ are $\mathbf{ZZ}, -\mathbf{ZI}$, and $-\mathbf{IZ}$. However, the vector $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ is not stabilized by any of these operators.

3.3. **Logical Paulis and encoding map for stabilizer codes.** In this section, we describe how to construct an encoding map (i.e., a canonical logical basis states) for a stabilizer code $\mathcal{C}(G)$. Since there are multiple possible choices of basis states for a subspace, the encoding map is not unique.

As before, let us assume that the stabilizer group is $G = \langle g_1, \dots, g_l \rangle$, where the generators $(g_i)_i$ are independent. Clearly, G is a normal subgroup of $N(G)$. To find a basis for the code $\mathcal{C}(G)$, we will first show that the quotient group $N(G)/G$ is isomorphic to the Pauli group on $n-l$ qubits, namely G_{n-l}.

Let us first count the number of elements in the group $N(G)/G$. An element h lies in $N(G)$ if and only if it commutes with all the generators of G. Hence, from the earlier discussion, $h \in N(G)$ if and only if

$$\Omega(g) J \Omega(h)^T = 0, \quad \forall g \in \{g_1, \dots, g_l\}.$$

Equivalently,

$$h \in N(G) \quad \iff \quad \Omega(G) J \Omega(h)^T = 0.$$

If we write $\Omega(G) = \big(G_1 \mid G_2\big)$, then this condition becomes

$$h \in N(G) \quad \iff \quad \Omega(G)' \, \Omega(h)^T = 0, \quad \text{where } \Omega(G)' = \big(G_2 \mid G_1\big).$$

Since the row rank of $\Omega(G)'$ is the same as that of $\Omega(G)$, namely l, the kernel of $\Omega(G)'$ is a $(2n-l)$-dimensional subspace. Hence, there are 2^{2n-l} possible vectors $\Omega(h)^T$. As each such $\Omega(h)^T$ corresponds to four different Pauli operators (up to global phases), the total number of elements in $N(G)$ is

$$4 \cdot 2^{2n-l} = 2^{2n+2-l} = 4^{\,n+1} \cdot 2^{-l}.$$

Therefore, the number of elements in $N(G)/G$ is

$$\frac{4^{\,n+1} \cdot 2^{-l}}{2^l} = 4^{(n-l)+1}.$$

Thus, the number of elements in $N(G)/G$ is exactly the same as the number of elements in G_{n-l}. Before writing down an explicit isomorphism between G_{n-l} and $N(G)/G$, we require the following exercise.

Exercise 3.11. ★ *Let $\{g_1, \dots, g_l\}$ be any independent set of elements in G_n. For any $c = (c_1, c_2, \dots, c_l) \in \mathbb{Z}_2^l$, consider the set of solutions*

$$g_j g = (-1)^{c_j} g g_j, \quad j = 1, \dots, l, \quad g \in G_n.$$

Show that this solution space is $(2n-l)$-dimensional.

Hint: Use the binary symplectic representation.

We are now in a position to define the isomorphism

$$\Psi : G_{n-l} \longrightarrow N(G)/G.$$

The Pauli group G_{n-l} has canonical generators

$$\{\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_{n-l}, \ \mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{n-l}\},$$

where $\mathbf{Z}_i$ and $\mathbf{X}_i$ denote the Pauli operators $\mathbf{Z}$ and $\mathbf{X}$ acting on the i-th qubit, respectively, and as the identity on all other qubits.

Since it suffices to define Ψ on this generating set, we introduce the images of the generators under Ψ:

$$\Psi(\mathbf{Z}_i) =: \overline{\mathbf{Z}}_i, \quad \Psi(\mathbf{X}_i) =: \overline{\mathbf{X}}_i, \quad 1 \le i \le n-l.$$

These operators $\overline{\mathbf{Z}}_i$ and $\overline{\mathbf{X}}_i$ are called the *logical Pauli operators*.
To define $\overline{\mathbf{Z}}_1$, let $\{g_1, \ldots, g_l\}$ be the generating elements of G, and take $c = (0, 0, \ldots, 0)$ in Exercise 3.11. This gives a $(2n-l)$-dimensional solution space of the equations

$$g_j g = g g_j, \quad j = 1, \ldots, l, \quad g \in G_n.$$

Of course, the independent generators $\{g_1, \ldots, g_l\}$ already belong to this solution space. We now choose some element $z_1 \in G_n$ that is independent of $\{g_1, \ldots, g_l\}$, take its equivalence class in $N(G)/G$, and denote it by $\overline{\mathbf{Z}}_1$.

To obtain $\overline{\mathbf{Z}}_2$, we repeat the same procedure with the enlarged set $\{g_1, \ldots, g_l, z_1\}$, again taking $c = (0, 0, \ldots, 0)$. This gives us a new independent solution, whose class in $N(G)/G$ we denote by $\overline{\mathbf{Z}}_2$. Proceeding inductively in this way, we define $\overline{\mathbf{Z}}_i$ for $1 \leq i \leq n-l$. Note that once $\overline{\mathbf{Z}}_{n-l}$ has been defined, the process cannot be continued further, since no additional linearly independent solution exists.

Next, we construct the elements $\overline{\mathbf{X}}_i$ for $1 \leq i \leq n-l$. To obtain $\overline{\mathbf{X}}_1$, consider the set $\{g_1, \ldots, g_l, z_1, z_2, \ldots, z_{n-l}\}$ and choose $c = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$ in Exercise 3.11, where the entry 1 is placed at the $(l+1)$-th position. By Exercise 3.11, the solution space is n-dimensional. None of these solutions lies in G, since all elements of G commute with $\overline{\mathbf{Z}}_1$. Choose one such solution $x_1 \in G_n$, take its equivalence class in $N(G)/G$, and denote it by $\overline{\mathbf{X}}_1$.

For $\overline{\mathbf{X}}_2$, we enlarge the set further to $\{g_1, \ldots, g_l, z_1, z_2, \ldots, z_{n-l}, x_1\}$, and take $c = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$, where the entry 1 is placed at the $(l+2)$-th position. This yields an $(n-1)$-dimensional solution space by Exercise 3.11. As before, we choose one solution x_2, take its class in $N(G)/G$, and denote it by $\overline{\mathbf{X}}_2$. Continuing this process, we obtain all $\overline{\mathbf{X}}_i$ for $1 \leq i \leq n-l$.

One readily verifies that the map Ψ thus defined can be extended to an isomorphism.

From now on, we will use $\overline{\mathbf{Z}}_i$ and z_i (as well as $\overline{\mathbf{X}}_i$ and x_i), interchangeably, for $1 \leq i \leq (n-l)$.

With the logical Pauli operators in hand, we can now define an encoding map as follows: for a bit string $(x_1, \ldots, x_{n-l}) \in \mathbb{Z}_2^{n-l}$, define the state

$$|x_1, \ldots, x_{n-l}\rangle_L \,^{5}$$

to be the unique (up to an overall phase) state with stabilizer group

$$\left\langle g_1, \ldots, g_l,\ (-1)^{x_1}\overline{\mathbf{Z}}_1, \ldots, (-1)^{x_{n-l}}\overline{\mathbf{Z}}_{n-l} \right\rangle.$$

Hence we obtain an encoding map for the stabilizer code. Note that this encoding map depends on the particular choices of $\overline{\mathbf{Z}}_i$ for $1 \leq i \leq (n-l)$.

Example 3.12. In the case of the bit-flip code, we considered $G \subseteq G_3$ generated by the two stabilizers **ZZ**I and I**ZZ**.

To find $\overline{\mathbf{Z}} = \overline{\mathbf{Z}}_1$, we apply Exercise 3.11. We need to solve the equation

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

[5] For a bit string $x \in \mathbb{Z}_2^{n-l}$, the notations $|x\rangle_L$ and $|x_L\rangle$ are used interchangeably; both denote the logical state $|x\rangle$.

One valid solution is $(0,0,0,1,1,1)$ for $(h_1,h_2,h_3,h_4,h_5,h_6)$. This solution is independent of $(1,1,0,0,0,0)$ and $(0,1,1,0,0,0)$, and corresponds to $\overline{\mathbf{Z}} = \mathbf{ZZZ}$.

The unique state stabilized by **ZZ**I, I**ZZ**, and **ZZZ** is $|000\rangle =: |0\rangle_L$. Similarly, the unique state stabilized by **ZZ**I, I**ZZ**, and $-$**ZZZ** is $|111\rangle =: |1\rangle_L$.

3.4. **CSS codes as stabilizer codes.** In this section, we will see how to construct quantum codes from classical codes using the stabilizer formalism.

Recall that a *linear code* is one whose encoding map is a linear map given by a vector space homomorphism

$$G : \mathbb{Z}_2^k \longrightarrow \mathbb{Z}_2^n,$$

where we assume that the columns of G are linearly independent.

Alternatively, a $[n,k]$ linear code can also be defined as the kernel of a *parity check matrix* H, which is an $(n-k)\times n$ matrix with linearly independent rows. In other words,

$$C = \{\, x \in \mathbb{Z}_2^n \;\mid\; Hx = 0 \,\}.$$

To see that these two definitions are equivalent:

- Given a parity check matrix H, pick k linearly independent vectors $y_1,\ldots,y_k$ spanning the kernel of H. Define

$$G = (y_1\ y_2\ \cdots\ y_k).$$

- Conversely, given a generator matrix G, one can construct a parity check matrix H by choosing $n-k$ linearly independent vectors $y_1,\ldots,y_{n-k}$ orthogonal to the columns of G. Then set

$$H = \begin{pmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_{n-k}^T \end{pmatrix}.$$

Here, orthogonality is understood with respect to the inner product modulo 2.

As an example, consider the *repetition code* with generator matrix

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

To construct a parity check matrix for this code, we choose the vectors $(1,1,0)$ and $(0,1,1)$, which are linearly independent and orthogonal to the column of G. Thus, we obtain

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

Here is a useful concept in the setting of linear codes: the *dual code*. If C is an $[n,k]$ linear code with generator matrix G and parity check matrix H, then the dual code $C^\perp$ is defined as the orthogonal complement of C. The dual code has generator matrix H^T and parity check matrix G^T.

A code C is called *weakly self-dual* if $C \subseteq C^\perp$, and *strictly self-dual* if $C = C^\perp$. It follows directly from the definition that C is weakly self-dual if and only if $G^T G = 0$.

Exercise 3.13. ★ *Consider the* $[7,4,3]$ *Hamming code* C*, whose parity check matrix is given by*

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

The generator matrix of C *is*

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

so the parity check matrix of $C^\perp$ *is* G^T*.*

Show that $C^\perp \subseteq C$ *by comparing the rows of* H *with linear combinations of the rows of* G^T*. Conclude that* $C^\perp$ *is weakly self-dual.*

Exercise 3.14. ★ *Let* C *be a linear code. Show that*

$$\sum_{y \in C} (-1)^{x \cdot y} = \begin{cases} |C| & \text{if } x \in C^\perp, \\ 0 & \text{if } x \notin C^\perp. \end{cases}$$

Let C_1 and C_2 be classical codes with parity-check matrices H_1 and H_2, satisfying the property that

$$C_2^\perp \subseteq C_1.$$

We then define the stabilizer code with check matrix

$$H = \begin{pmatrix} H_1 & 0 \\ 0 & H_2 \end{pmatrix}.$$

We now show that H defines a valid stabilizer code. Consider arbitrary rows $(h_1 \;\; 0) \in (H_1 \mid 0)$ and $(0 \;\; h_2) \in (0 \mid H_2)$. We need to verify that

$$(h_1 \;\; 0) J (0 \;\; h_2)^T = 0,$$

which is equivalent to showing $h_1 h_2^T = 0$.

This condition holds if h_2^T is a codeword of C_1, i.e., if $h_2^T \in \ker(H_1)$. By assumption, $\ker(H_2)^\perp \subseteq C_1$. Thus, it suffices to show that $h_2^T \in \ker(H_2)^\perp$.

Let h^T be a codeword of C_2, i.e., $h^T \in \ker(H_2)$. Then $h_2 h^T = 0$, which implies $h_2^T \in \ker(H_2)^\perp$. Hence, the commutation condition is satisfied, and H indeed defines a valid stabilizer code.

We denote the above CSS code by $\mathrm{CSS}(H_1, H_2)$ or equivalently by $\mathrm{CSS}(C_1, C_2)$. Next, we determine the exact distance of the CSS code $\mathrm{CSS}(H_1, H_2)$.

Proposition 3.15. *Let* C_1 *and* C_2 *be classical codes with parity-check matrices* H_1 *and* H_2*, such that* $C_2^\perp \subseteq C_1$*. Define*

$$d_1 = \min\{\, \mathrm{wt}(v) \;\mid\; v \in C_1 \setminus C_2^\perp \,\}, \qquad d_2 = \min\{\, \mathrm{wt}(v) \;\mid\; v \in C_2 \setminus C_1^\perp \,\}.$$

Then the distance d *of the CSS code* $\mathrm{CSS}(H_1, H_2)$ *is*

$$d = \min\{d_1, d_2\}.$$

Proof. Let $v_1 \in C_1 \setminus C_2^\perp$, and put $c_1 = (0, v_1)$. Since

$$v_1 \notin C_2^\perp = (\ker H_2)^\perp = \mathrm{Im}(H_2^T),$$

it follows that v_1 is not a linear combination of the rows of H_2. Hence c_1 is not a linear combination of the rows of

$$H = \begin{pmatrix} H_1 & 0 \\ 0 & H_2 \end{pmatrix}.$$

This means there is no $g \in G$ (the stabilizer group) with $\Omega(g) = c_1$. In other words, $c_1 \notin \Omega(G)$.

On the other hand, $c_1 \in \Omega(N(G))$: indeed, for any row $(h_1 \;\; 0) \in (H_1 \mid 0)$, we have

$$(0 \;\; v_1) J (h_1 \;\; 0)^T = 0,$$

which follows since $v_1 \in C_1$. Thus, $c_1 \in \Omega(N(G)) \setminus \Omega(G)$. The symplectic weight of c_1 is

$$\mathrm{wt}_s(c_1) = \mathrm{wt}(v_1).$$

By Equation 3.6, this shows $d \leq d_1$.

A similar argument with $v_2 \in C_2 \setminus C_1^\perp$ and $c_2 = (v_2, 0)$ shows that $d \leq d_2$. Hence

$$d \leq \min\{d_1, d_2\}.$$

Now let $c = (v_1, v_2) \in \Omega(N(G)) \setminus \Omega(G)$. Since $c \notin \Omega(G)$, at least one of the following must hold: $v_1 \notin C_1^\perp$, or $v_2 \notin C_2^\perp$.

Suppose $v_2 \notin C_2^\perp$. Since $c \in \Omega(N(G))$, we have $H_1 v_2 = 0$, i.e., $v_2 \in C_1$. Thus $v_2 \in C_1 \setminus C_2^\perp$. Similarly, if $v_1 \notin C_1^\perp$, then $v_1 \in C_2 \setminus C_1^\perp$.

In either case, we have

$$\mathrm{wt}_s(c) \geq \min\{\mathrm{wt}(v_1), \mathrm{wt}(v_2)\} \geq \min\{d_1, d_2\}.$$

Hence $d \geq \min\{d_1, d_2\}$. Combining both inequalities, we conclude

$$d = \min\{d_1, d_2\}.$$

□

Notes

- Most of the material presented above is drawn from Sections 10.4 and 10.5 of Nielsen and Chuang 2010.
- Readers are also encouraged to consult the thesis of Gottesman 1997, in which the concept of stabilizer codes was first introduced.

References for Lecture 5 & 6.

Gottesman, Daniel (1997). *Stabilizer Codes and Quantum Error Correction*. eprint: arXiv:quant-ph/9705052.

Nielsen, Michael A. and Isaac L. Chuang (2010). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press.

Lecture 7 & 8: Toric codes, HGP codes

By an X-type operator, we mean a Pauli operator whose entries are either I or $\mathbf{X}$. Similarly, one can define a Z-type operator.

In this section, we introduce the toric code and a class of quantum codes that can be constructed as a "product" of two classical codes, known as the *hypergraph product codes*.

Before discussing the toric code, we first present some *visual representations* of codes that were introduced earlier.

Let us begin with a general setup. Let (V, E, F) be a planar graph, where V is the set of vertices, E is the set of edges, and F is the set of faces. We aim to define a stabilizer code based on this data.

Suppose we place a qubit on each edge $e \in E$. Then we can define an X-type operator $\mathbf{X}_v$ (or a Z-type operator $\mathbf{Z}_v$) acting on all edges incident to a vertex $v \in V$. Similarly, we define $\mathbf{X}_f$ (or $\mathbf{Z}_f$) acting on all edges incident to a face $f \in F$. Alternatively, one could place qubits on vertices and define $\mathbf{X}_f$ (and $\mathbf{Z}_f$) for $f \in F$, or $\mathbf{X}_e$ (and $\mathbf{Z}_e$) for $e \in E$.

Figure 1

In what follows, we will define quantum codes in this manner—by placing qubits (green dots) on vertices or edges and defining stabilizer generators through such X-type and Z-type operators.

Let us begin with the bit-flip code. Recall that it is a three-qubit code with stabilizer generators **ZZ**I and I**ZZ**. To obtain a visual representation of this code, consider the graph (see Figure 1) with three vertices and two edges.

Now, place the qubits on the vertices, and associate a Z-type stabilizer generator with each edge. The stabilizer code defined by these generators is precisely the bit-flip code.

One can also give a visual representation of the $[[4, 1, 2]]$ code, as shown in Figure 2. Note that the qubits are placed on the four vertices. For the face f, we have an X-type stabilizer generator, while for the two edges as in the figure, we have Z-type stabilizer generators.

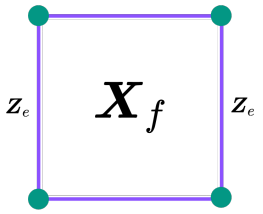
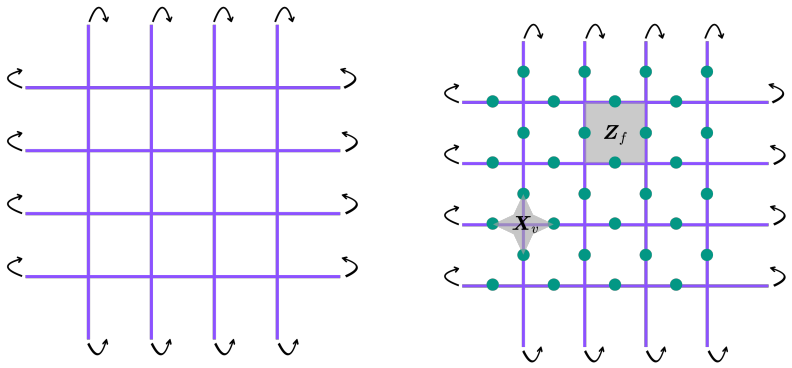


Figure 2

3.5. **Toric code.** We now turn to the definition of the toric code. Fix a number $L > 1$. Imagine an $L \times L$ grid with periodic boundary conditions—that is, the grid is wrapped around to form a torus. See Figure 3 (A) for the case $L = 4$. Although this graph is not planar, it defines a vertex set V, an edge set E, and a face set F. A simple counting argument shows that there are L^2 vertices, $2L^2$ edges, and L^2 faces.

To define a stabilizer code, we place the qubits on the edges. For each vertex $v \in V$, we define an X-type stabilizer generator $\mathbf{X}_v$, and for each face $f \in F$, we define a Z-type stabilizer generator $\mathbf{Z}_f$. (See Figure 3 (B).) One can easily verify that these stabilizers commute. Hence, this construction defines a valid stabilizer code, known as the *toric code*. The stabilizer group will be denoted by G, as usual.

The following exercise shows that the stabilizer generators are not independent.



(A) (B)

FIGURE 3. Toric code

Exercise 3.16. ★ *Show that*

$$\prod_{v\in V} \mathbf{X}_v = \prod_{f\in F} \mathbf{Z}_f = I.$$

Also, show that one must remove exactly one $\mathbf{X}_v$ *and exactly one* $\mathbf{Z}_f$ *from the set of stabilizer generators to make the remaining ones independent. Hence, conclude that there are two logical qubits in the toric code, i.e., the codespace dimension is* 4.

Before computing the distance of the toric code, let us first show that the toric code is *symmetric*—that is, if we exchange $\mathbf{X}$ and $\mathbf{Z}$, the codespace remains unchanged, although this symmetry is not immediately apparent from the definition.

To see this, draw a dotted grid passing through the qubits of the toric code, as shown in Figure 5 (A) below. Now rotate this dotted grid by 45° in the anticlockwise direction. If we erase the original grid and keep only the rotated one—while remembering the locations of the stabilizer generators—the resulting image (Figure 5 (B)) provides an alternative representation of the toric code. In this new picture, the qubits are placed on the vertices, and the faces alternately define X-type and Z-type stabilizer generators, as illustrated in Figure 5 (B). In this representation, the symmetry of the toric code becomes evident. Consequently, the X-distance of the toric code is equal to its Z-distance. [6]

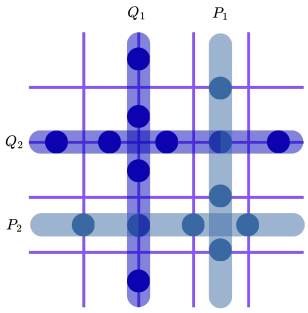


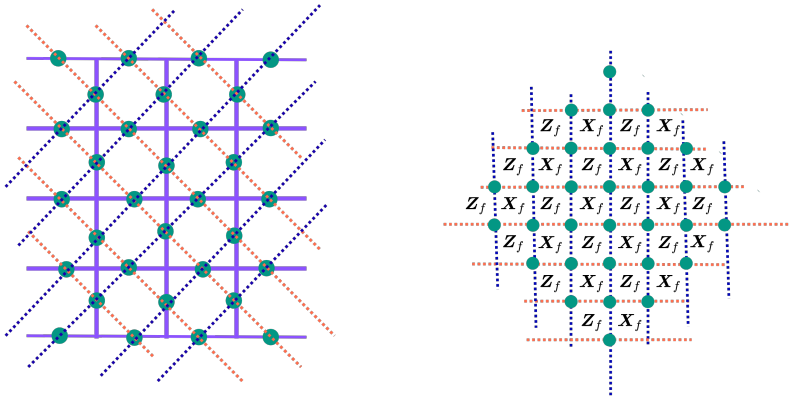
FIGURE 4

Let us define the X-type operators P_1 and P_2, and the Z-type operators Q_1 and Q_2, as illustrated in Figure 4.

Note that $P_1, P_2, Q_1, Q_2 \in N(G)$. From the diagram, the following commutation and anticommutation relations are straightforward to verify:

$$P_1P_2 = P_2P_1, \quad Q_1Q_2 = Q_2Q_1, \quad P_1Q_1 = Q_1P_1, \quad P_2Q_2 = Q_2P_2,$$

[6]The X-distance (resp. Z-distance) is the minimum weight of an undetectable X-type (resp. Z-type) operator.



(A) (B)

FIGURE 5. Symmetry of the toric code

$$P_1Q_2 = -Q_2P_1, \quad P_2Q_1 = -Q_1P_2.$$

We now show that P_1, P_2, Q_1, and Q_2 lie in $N(G) \setminus G$. Suppose, for contradiction, that $P_1 \in G$. Then $P_1, Q_2 = -Q_2P_1$ would imply that Q_2 anticommutes with an element of G, contradicting the fact that $Q_2 \in N(G)$. The same argument applies to P_2, Q_1, and Q_2.

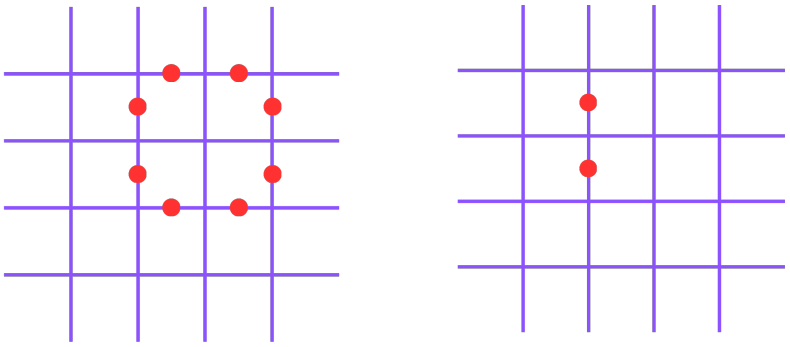
Hence, these operators represent nontrivial logical operators. Taking the corresponding equivalence classes, we obtain

$$[P_1] := \overline{\mathbf{X}}_1, \qquad [P_2] := \overline{\mathbf{X}}_2, \qquad [Q_1] := \overline{\mathbf{Z}}_2, \qquad [Q_2] := \overline{\mathbf{Z}}_1.$$

3.6. **Distance of the toric code.** By symmetry, it suffices to consider only Z-type errors. Assume we have a subset $T \subseteq E$ of edges on which errors occur. Then T can be decomposed as a union of elements belonging to the following three categories:

- **Contractible loops:** These are loops (see Figure 6 (A)) that can be continuously shrunk to a point on the torus. The corresponding Z-type operators belong to the stabilizer group G.
- **Open walks with two distinct endpoints:** In this case, the edges in the walk are not repeated, and the degree of each vertex is at most 2, while the endpoint vertices have degree 1. (See Figure 6 (B).) The Z-type operators corresponding to such walks anticommute with the X-type stabilizers of the endpoint vertices. Hence, these Z-type operators are not elements of $N(G)$ and represent *detectable errors*.
- **Non-contractible loops:** These are loops that cannot be continuously deformed to a point on the torus. Typical examples include the operators P_1, P_2, Q_1, and Q_2. These are logical operators, each having weight at least L.

From the above, we conclude that the distance of the toric code is L, since there exist logical operators (for instance, P_1, P_2, Q_1, and Q_2) of weight L. Hence the toric code is a $[[2L^2, 2, L]]$ code.



(A) (B)

FIGURE 6

3.7. **Surface code.** We now consider the toric code *without* periodic boundary conditions. To do this, fix a number $L > 1$. Consider an $(L-1) \times (L-1)$ rectangular grid with $L-1$ edges on each side. Identify the vertical (left and right) boundaries to obtain a cylindrical surface. Now, perform a vertical cut located between any two adjacent vertical lines of the lattice.

When this cut surface is mapped back onto the plane, the resulting grid defines an $L \times L$ *surface code* with *smooth* (top and bottom) boundaries and *rough* (left and right) boundaries. See Figure 7 for an illustration.

The above grid again defines a vertex set V, an edge set E, and a face set F. Note that the open, dangling parts on the left and right boundaries do not define vertices, while the open faces on the left and right are still counted as elements of F. As in the toric code, place the qubits on the edges, X-type stabilizers on the vertices, and Z-type stabilizers on the faces. The resulting code is known as the *surface code*. The computation of its parameters is left as an exercise below.

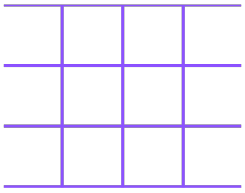


FIGURE 7

Exercise 3.17. ★ *Show that the above surface code is a* $[[\, L^2 + (L-1)^2,\ 1,\ L \,]]$ *quantum code.*

Exercise 3.18. ★ ***(Rotated surface code)***

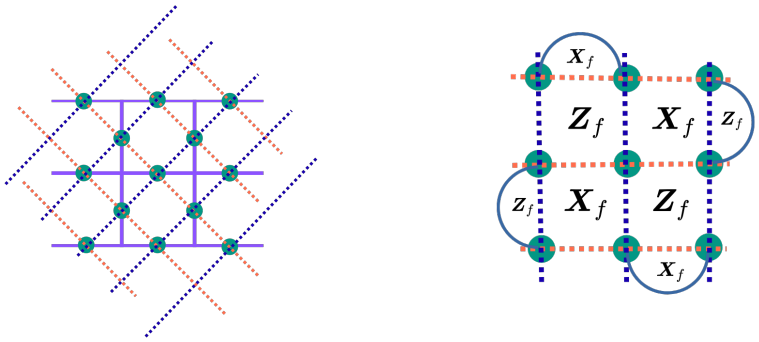
Consider the surface-code lattice for $L = 3$ *(qubits on edges;* X*-stabilizers on vertices;* Z*-stabilizers on faces).*

Draw dotted lines passing through the qubits as in the symmetry argument for the toric code, rotate the dotted grid by $45°$ *(Figure 8 (A)), erase the original grid while keeping track of the stabilizer generators, remove the four corner vertices (qubits), and remove appropriate stabilizer generators as indicated in (Figure 8 (B)).*

(1) Show that the resulting rotated lattice contains 9 *physical qubits, has* 8 *independent stabilizer generators, and therefore encodes* $k = 1$ *logical qubit.*

(2) Prove that the minimum weight of a nontrivial logical operator is 3*, hence the code has distance* $d = 3$*. (Hint: identify the shortest noncontractible paths/loops on the rotated lattice.)*

(3) Prove that this $[[9, 1, 3]]$ *rotated surface code is* not *Shor's code.*



(A) (B)

FIGURE 8. [[9,1,3]] rotated surface code

3.8. **Hypergraph product code.** A Tanner graph is a convenient graphical representation of a parity-check matrix H of a linear code C. We now describe its construction.

A Tanner (or factor) graph associated with $H = (H_{ij})$ has two types of vertices: *check nodes* and *variable nodes*. For each row i of H, we include a check node, and for each column j, we include a variable node. An edge connects check node i to variable node j whenever $H_{ij} = 1$.

The following figure (Figure 9) illustrates the Tanner graph corresponding to the parity-check matrix

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

is

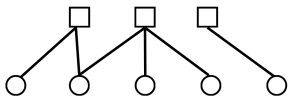
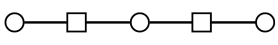


FIGURE 9

Similarly, the Tanner graph for the parity-check matrix of the repetition code

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

is



Hence, the Tanner graph associated with a classical code C is a bipartite graph $\mathscr{G} = ((\mathscr{V}, \mathscr{C}), \mathscr{E})$, where $\mathscr{V}$ denotes the set of variable nodes, $\mathscr{C}$ denotes the set of check nodes, and $\mathscr{E}$ represents the set of edges.

We now define the Tanner graph for CSS codes. Recall that a check matrix of a CSS code can be written as

$$H = \begin{pmatrix} H_X & 0 \\ 0 & H_Z \end{pmatrix},$$

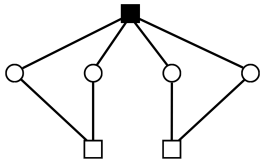
where $H_X \in \mathrm{M}_{m_x \times n}(\mathbb{Z}_2)$, $H_Z \in \mathrm{M}_{m_z \times n}(\mathbb{Z}_2)$, and $H \in \mathrm{M}_{m \times n}(\mathbb{Z}_2)$ with $m = m_x + m_z$, for some integers m_x and m_z. We may view H_X and H_Z as the parity-check matrices of two classical linear codes.

Since a CSS code is constructed from two classical codes defined by H_X and H_Z, we can form its Tanner graph by combining the Tanner graphs of these two component codes. The resulting graph is called a *CSS Tanner graph*. It is clear that a CSS Tanner graph contains three types of vertices: $\mathscr{V}$ (corresponding to the columns of H_X and H_Z), $\mathscr{C}_X$ (corresponding to the rows of H_X), and $\mathscr{C}_Z$ (corresponding to the rows of H_Z). We denote the entire structure by $((\mathscr{V}, \mathscr{C}_X, \mathscr{C}_Z), \mathscr{E})$.

We illustrate this with the following example. Consider the CSS code with

$$H_X = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}, \quad H_Z = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

The corresponding CSS Tanner graph is shown below.



Note that the check nodes corresponding to H_X are colored black to distinguish them from the nodes of H_Z.

Before defining hypergraph product codes, we first introduce the tensor product of two classical codes. Let C_1 and C_2 be two classical linear codes with generator matrices G_1 and G_2, and parity-check matrices H_1 and H_2, respectively. The tensor product code $C_1 \otimes C_2$ consists of all codewords that are tensor products of codewords from C_1 and C_2.

Exercise 3.19. ★ *Assume that G_1 is an $n_1 \times k_1$ full-rank matrix and G_2 is an $n_2 \times k_2$ full-rank matrix. Show that the generator matrix G and the parity-check matrix H of the tensor product code $C_1 \otimes C_2$ are given by*

$$G = G_1 \otimes G_2, \qquad H = \begin{pmatrix} \mathrm{I}_{n_1} \otimes H_2 \\ H_1 \otimes \mathrm{I}_{n_2} \end{pmatrix}.$$ [7]

Assume C_1 has distance d_1 and C_2 has distance d_2. Show that the distance of $C_1 \otimes C_2$ is $d_1 d_2$.

Let us now introduce the concept of the product of two graphs. Given two graphs $\mathscr{G}_1 = (\mathscr{V}_1, \mathscr{E}_1)$ and $\mathscr{G}_2 = (\mathscr{V}_2, \mathscr{E}_2)$, their product $\mathscr{G}_1 \times \mathscr{G}_2$ is defined as the graph with vertex set

$$\mathscr{V} = \{(x, y) \mid x \in \mathscr{V}_1,\ y \in \mathscr{V}_2\}.$$

[7] I_r denotes the identity matrix of size $r \times r$.

For $(x,y),(x',y') \in \mathscr{V}$, there is an edge between them if and only if either $x = x'$ and $\{y,y'\} \in \mathscr{E}_2$, or $y = y'$ and $\{x,x'\} \in \mathscr{E}_1$. Here, by $\{x,x'\} \in \mathscr{E}_1$ we mean that there is an edge between x and x' in $\mathscr{E}_1$, and similarly for $\{y,y'\} \in \mathscr{E}_2$.

Exercise 3.20. ★ *Draw the product of two graphs, each being the Tanner graph of the repetition code.*

Definition 3.21 (Hypergraph product codes)**.** *Let $\mathscr{G}_1 = ((\mathscr{V}_1,\mathscr{C}_1),\mathscr{E}_1)$ and $\mathscr{G}_2 = ((\mathscr{V}_2,\mathscr{C}_2),\mathscr{E}_2)$ be two Tanner graphs. The* hypergraph product code*, denoted by* $\mathrm{HGP}(\mathscr{G}_1,\mathscr{G}_2)$*, is the CSS code with the CSS Tanner graph*

$$\mathscr{G} = \mathscr{G}_1 \times \mathscr{G}_2 = ((\mathscr{V},\mathscr{C}_X,\mathscr{C}_Z),\mathscr{E}),$$

defined as

$$\mathscr{V} = (\mathscr{V}_1 \times \mathscr{V}_2) \cup (\mathscr{C}_1 \times \mathscr{C}_2), \qquad \mathscr{C}_X = \mathscr{C}_1 \times \mathscr{V}_2, \qquad \mathscr{C}_Z = \mathscr{V}_1 \times \mathscr{C}_2.$$

If $\mathscr{G}_1 = ((\mathscr{V}_1,\mathscr{C}_1),\mathscr{E}_1)$ and $\mathscr{G}_2 = ((\mathscr{V}_2,\mathscr{C}_2),\mathscr{E}_2)$ are the Tanner graphs associated with linear codes having parity-check matrices H_1 and H_2, respectively, we also denote the corresponding hypergraph product code by $\mathrm{HGP}(H_1,H_2)$.

Proposition 3.22. *Let $\mathscr{G}_1 = ((\mathscr{V}_1,\mathscr{C}_1),\mathscr{E}_1)$ and $\mathscr{G}_2 = ((\mathscr{V}_2,\mathscr{C}_2),\mathscr{E}_2)$ be two Tanner graphs. Then $\mathscr{G} = \mathscr{G}_1 \times \mathscr{G}_2 = ((\mathscr{V},\mathscr{C}_X,\mathscr{C}_Z),\mathscr{E})$, as defined in Definition 3.21, is indeed a CSS Tanner graph.*

Proof. We need to show that $\mathscr{G} = \mathscr{G}_1 \times \mathscr{G}_2 = ((\mathscr{V},\mathscr{C}_X,\mathscr{C}_Z),\mathscr{E})$ arises from a valid CSS code. For a graph $(\mathscr{V},\mathscr{E})$ and a vertex $v \in \mathscr{V}$, define its neighbourhood by

$$N(v) := \{\, u \in \mathscr{V} \mid \{u,v\} \in \mathscr{E} \,\}.$$

Let $(c_1,v_2) \in \mathscr{C}_X$ and $(v_1,c_2) \in \mathscr{C}_Z$. From the definition of CSS codes, it suffices to show that the intersection $N((c_1,v_2)) \cap N((v_1,c_2))$ contains an even number of vertices.

Let $(a,b) \in N((c_1,v_2)) \cap N((v_1,c_2))$. We analyse the possible cases:

Case 1: $a = c_1$, $\{b,v_2\} \in \mathscr{E}_2$, and $a = v_1$, $\{b,c_2\} \in \mathscr{E}_2$. This is impossible since $\{b,v_2\} \in \mathscr{E}_2$ contradicts $\{b,c_2\} \in \mathscr{E}_2$.

Case 2: $a = c_1$, $\{b,v_2\} \in \mathscr{E}_2$, and $b = c_2$, $\{a,v_1\} \in \mathscr{E}_1$. This implies $\{c_2,v_2\} \in \mathscr{E}_2$ and $\{c_1,v_1\} \in \mathscr{E}_1$, giving $(a,b) = (c_1,c_2)$.

Case 3: $\{a,c_1\} \in \mathscr{E}_1$, $b = v_2$, and $a = v_1$, $\{b,c_2\} \in \mathscr{E}_2$. This again implies $\{c_2,v_2\} \in \mathscr{E}_2$ and $\{c_1,v_1\} \in \mathscr{E}_1$, giving $(a,b) = (v_1,v_2)$.

Case 4: $\{a,c_1\} \in \mathscr{E}_1$, $b = v_2$, and $b = c_2$, $\{a,v_1\} \in \mathscr{E}_1$. This is also impossible, just as in Case 1.

Considering the above, whenever $\{c_2,v_2\} \in \mathscr{E}_2$ and $\{c_1,v_1\} \in \mathscr{E}_1$, we have $(c_1,c_2),(v_1,v_2) \in N((c_1,v_2)) \cap N((v_1,c_2))$. Hence, the intersection is either empty or contains an even number of vertices. This completes the proof. □

In the following we will derive the number of encoded qubits, and the distance of hypergraph product codes.

For a linear code C with parity-check matrix H, we define the transpose code C^T that has the parity-check matrix H^T. We start with a small exercise.

Exercise 3.23. ★ *The dimension of C^T is $k^T = k - n + m$, where k is the dimension of C, n the number of physical bits (columns) in C and m the number of rows in H.*

It follows from the above lemma that if H is full rank then $n-m=k$ which implies that $k^T=0$. In this case the distance d^T of C^T is defined to be ∞. Note that the Tanner graph of C and the Tanner graph C^T look the same except the variable and check nodes are interchanged. For a Tanner graph $\mathscr{G}$ associated to a classical code C, we denote by $\mathscr{G}^T$, the Tanner graph associated to the classical code C^T.

Let $\mathscr{G}_1=((\mathscr{V}_1,\mathscr{C}_1),\mathscr{E}_1)$ and $\mathscr{G}_2=((\mathscr{V}_2,\mathscr{C}_2),\mathscr{E}_2)$ be Tanner graphs of two classical codes C_1 and C_2, respectively. Recall that the hypergraph product code has the CSS Tanner graph $\mathscr{G}=\mathscr{G}_1\times\mathscr{G}_2=((\mathscr{V},\mathscr{C}_X,\mathscr{C}_Z),\mathscr{E})$ given by:

$$\mathscr{V}=\mathscr{V}_1\times\mathscr{V}_2\cup\mathscr{C}_1\times\mathscr{C}_2,\quad \mathscr{C}_X=\mathscr{C}_1\times\mathscr{V}_2,\quad \mathscr{C}_Z=\mathscr{V}_1\times\mathscr{C}_2.$$

Let us consider the Tanner sub-graph $\mathscr{G}_1\otimes\mathscr{G}_2$ of $\mathscr{G}_1\times\mathscr{G}_2$ generated by vertices of variable nodes $\mathscr{V}_1\times\mathscr{V}_2$ and check nodes $\mathscr{C}_1\times\mathscr{V}_2\cup\mathscr{V}_1\times\mathscr{C}_2$. Also consider the Tanner sub-graph $\mathscr{G}_1\otimes_X\mathscr{G}_2$ of $\mathscr{G}_1\times\mathscr{G}_2$ generated by variable nodes $\mathscr{V}$ and check nodes $\mathscr{C}_X$ and the Tanner sub-graph $\mathscr{G}_1\otimes_Z\mathscr{G}_2$ of $\mathscr{G}_1\times\mathscr{G}_2$ generated by variable nodes $\mathscr{V}$ and check nodes $\mathscr{C}_Z$.

Exercise 3.24. ★ *Show that the Tanner graph $\mathscr{G}_1\otimes\mathscr{G}_2$ is given by the classical code $C_1\otimes C_2$. Also show that the Tanner sub-graph $\mathscr{G}_1\otimes_X\mathscr{G}_2$ is exactly the Tanner graph $(\mathscr{G}_1^T\otimes\mathscr{G}_2)^T$, the Tanner sub-graph $\mathscr{G}_1\otimes_Z\mathscr{G}_2$ is exactly the Tanner graph $(\mathscr{G}_1\otimes\mathscr{G}_2^T)^T$,*

We can also write the CSS code $\mathrm{HGP}(H_1,H_2)$ as a check matrix using the parity check matrices H_1 and H_2 of the classical Tanner graphs $\mathscr{G}_1=((\mathscr{V}_1,\mathscr{C}_1),\mathscr{E}_1)$ and $\mathscr{G}_2=(\mathscr{V}_2,\mathscr{C}_2,\mathscr{E}_2)$. Assume $H_1\in\mathrm{M}_{\mathrm{m}_1\times\mathrm{n}_1}(\mathbb{Z}_2)$ and $H_2\in\mathrm{M}_{\mathrm{m}_2\times\mathrm{n}_2}(\mathbb{Z}_2)$. Then the following exercise shows what the check matrix

$$\left(\begin{array}{c|c} H_X & 0 \\ 0 & H_Z \end{array}\right),$$

of $\mathrm{HGP}(H_1,H_2)$ looks like in terms of H_1 and H_2.

Exercise 3.25. ★

$$\left(\begin{array}{c|c} H_X & 0 \\ 0 & H_Z \end{array}\right),$$

is the check matrix $\mathrm{HGP}(\mathrm{G}_1,\mathrm{G}_2)$ *where*

$$H_X=\left(\begin{array}{c|c} H_1\otimes\mathrm{I}_{n_2} & \mathrm{I}_{m_1}\otimes H_2^T \end{array}\right),\quad H_Z=\left(\begin{array}{c|c} \mathrm{I}_{n_1}\otimes H_2 & H_1^T\otimes\mathrm{I}_{m_2} \end{array}\right).$$

Give an alternative way to prove Proposition 3.22.

Now we have the following result regarding the number of encoded qubits of a hypergraph product code.

Theorem 3.26. *Let $H_1\in\mathrm{M}_{\mathrm{m}_1\times\mathrm{n}_1}(\mathbb{Z}_2)$ and $H_2\in\mathrm{M}_{\mathrm{m}_2\times\mathrm{n}_2}(\mathbb{Z}_2)$ be two party-check matrices of two linear codes C_1 and C_2 with parameters $[n_1,k_1,d_1]$, $[n_2,k_2,d_2]$, respectively. Then the number of encodes qubits in* $\mathrm{HGP}(H_1,H_2)$ *is*

$$k=k_1k_2+k_1^Tk_2^T.$$

Proof. We know that

$$k=n_1n_2+m_1m_2-\mathrm{Rank}(H_X)-\mathrm{Rank}(H_Z).$$

Let $\mathscr{G}_1=((\mathscr{V}_1,\mathscr{C}_1),\mathscr{E}_1)$ and $\mathscr{G}_2=((\mathscr{V}_2,\mathscr{C}_2),\mathscr{E}_2)$ be the Tanner graphs corresponding to H_1 and H_2. Denote the classical code corresponding to the Tanner (sub)graph $\mathscr{G}_1\otimes_X\mathscr{G}_2$ by C_X, and the code for $\mathscr{G}_1\otimes_Z\mathscr{G}_2$ by C_Z. Then it is clear that $\dim(C_X)=n_1n_2+m_1m_2-\mathrm{Rank}(H_X)$ and $\dim(C_Z)=n_1n_2+m_1m_2-\mathrm{Rank}(H_Z)$. We have

$$k = \dim(C_X) + \dim(C_Z) - (n_1 n_2 + m_1 m_2). \tag{3.7}$$

Using Exercise 3.23, we have

$$\dim((C_X)^T) = \dim(C_X) - (n_1 n_2 + m_1 m_2) + m_1 n_2.$$
$$\dim((C_Z)^T) = \dim(C_Z) - (n_1 n_2 + m_1 m_2) + n_1 m_2.$$

Now use Exercise 3.24 to get

$$\dim(C_X) = n_1 n_2 + m_1 m_2 - m_1 n_2 + k_1^T k_2.$$

Similarly,

$$\dim(C_Z) = n_1 n_2 + m_1 m_2 - n_1 m_2 + k_1 k_2^T.$$

Hence using Equation 3.7,

$$\begin{aligned} k =& n_1 n_2 + m_1 m_2 - m_1 n_2 + k_1^T k_2 - n_1 m_2 + k_1 k_2^T \\ =& n_1 (n_2 - m_2) - m_1 (n_2 - m_2) + k_1^T k_2 + k_1 k_2^T \\ =& (n_1 - m_1)(n_2 - m_2) + k_1^T k_2 + k_1 k_2^T \\ =& \left(k_1 - k_1^T\right)\left(k_2 - k_2^T\right) + k_1^T k_2 + k_1 k_2^T \text{ (using Exercise 3.24)} \\ =& k_1 k_2 + k_1^T k_2^T \end{aligned}$$

□

Theorem 3.27. *Let $H_1 \in \mathrm{M}_{m_1 \times n_1}(\mathbb{Z}_2)$ and $H_2 \in \mathrm{M}_{m_2 \times n_2}(\mathbb{Z}_2)$ be the parity-check matrices of two linear codes C_1 and C_2 with parameters $[n_1, k_1, d_1]$ and $[n_2, k_2, d_2]$, respectively. Then the distance d of $\mathrm{HGP}(H_1, H_2)$ satisfies*

$$d \geq \min(d_1, d_2, d_1^T, d_2^T).$$

Proof. Let E be an element of $N(G)$ such that $\mathrm{wt}(E) < \min(d_1, d_2, d_1^T, d_2^T)$. We will show that E is in fact an element of the stabilizer group G.

We can always write $E = E_Z \cdot E_X$, where E_Z is a Z-type operator and E_X is an X-type operator, satisfying $\mathrm{wt}(E_Z) \leq \mathrm{wt}(E)$ and $\mathrm{wt}(E_X) \leq \mathrm{wt}(E)$. We first show that $E_Z \in G$.

The operator E_Z clearly commutes with all the X-type stabilizers of the code $\mathrm{HGP}(H_1, H_2)$. As before, denote by C_X the classical code corresponding to H_X, and by C_Z the code corresponding to H_Z. Using the binary symplectic representation, E_Z defines a codeword z of C_X. Since $\mathrm{wt}(E_Z) \leq \mathrm{wt}(E)$, we have $\mathrm{wt}(z) \leq \min(d_1, d_2^T)$.

If we denote the Tanner graph of H_1 by $((\mathscr{V}_1, \mathscr{C}_1), \mathscr{E}_1)$ and that of H_2 by $((\mathscr{V}_2, \mathscr{C}_2), \mathscr{E}_2)$, then the support of z satisfies

$$\mathrm{Supp}(z) \subseteq (\mathscr{V}_1 \times \mathscr{V}_2) \cup (\mathscr{C}_1 \times \mathscr{C}_2),$$

where $\mathrm{Supp}(z)$ denotes the usual support of a function. Define the following subsets:

$$\mathscr{V}_1' := \{\, v' \in \mathscr{V}_1 \mid \exists\, v \in \mathscr{V}_2,\ (v', v) \in \mathrm{Supp}(z) \,\}, \qquad \mathscr{C}_2' := \{\, c' \in \mathscr{C}_2 \mid \exists\, c \in \mathscr{C}_1,\ (c, c') \in \mathrm{Supp}(z) \,\}.$$

Let $\mathscr{G}_1'$ denote the Tanner subgraph of $((\mathscr{V}_1, \mathscr{C}_1), \mathscr{E}_1)$ induced by the vertex set $(\mathscr{V}_1', \mathscr{C}_1)$, and let $\mathscr{G}_2'$ denote the Tanner subgraph of $((\mathscr{V}_2, \mathscr{C}_2), \mathscr{E}_2)$ induced by $(\mathscr{V}_2, \mathscr{C}_2')$. Let C_i' denote the linear code defined by $\mathscr{G}_i'$, for $i = 1, 2$.

Since the number of elements in $\mathscr{V}_1'$ is less than d_1 (because $\mathrm{wt}(z) < d_1$), the dimension of C_1' must be zero. Indeed, any nonzero codeword of weight less than d_1 would yield a nonzero codeword of C_1 by padding with zeros, contradicting that the minimum distance of C_1 is d_1.

Similarly, the dual code $C_2'^T$ has zero dimension using $\mathscr{G}_2'$. Then, by Theorem 3.26, we have $\mathrm{HGP}(\mathscr{G}_1', \mathscr{G}_2') = 0$.

Let z' denote the restriction of z to

$$(\mathscr{V}_1' \times \mathscr{V}_2) \cup (\mathscr{C}_1 \times \mathscr{C}_2').$$

Denote by $h_Z(v_1, c_2)$ the row of H_Z corresponding to the Z-check $(v_1, c_2) \in \mathscr{V}_1 \times \mathscr{C}_2$, and by $h'_Z(v_1', c_2')$ the analogous row in $\mathrm{HGP}(\mathscr{G}_1', \mathscr{G}_2')$ for $(v_1', c_2') \in \mathscr{V}_1' \times \mathscr{C}_2'$.

Since $\mathrm{HGP}(\mathscr{G}_1', \mathscr{G}_2') = 0$, the operator corresponding to z' is a Z-type stabilizer of $\mathrm{HGP}(\mathscr{G}_1', \mathscr{G}_2')$, and hence

$$z' = \bigoplus_{(v_1', c_2') \in J} h'_Z(v_1', c_2'),$$

for some subset $J \subseteq \mathscr{V}_1' \times \mathscr{C}_2'$.

Consider the neighbourhood of $(v_1', c_2') \in \mathscr{V}_1' \times \mathscr{C}_2'$ in $\mathscr{G}_1' \times \mathscr{G}_2'$, namely

$$N' := \{\, x \in (\mathscr{V}_1' \times \mathscr{V}_2) \cup (\mathscr{C}_1 \times \mathscr{C}_2') \;\mid\; x \text{ is connected to } (v_1', c_2') \text{ by an edge in } \mathscr{G}_1' \times \mathscr{G}_2' \,\},$$

and the neighbourhood of $(v_1', c_2') \in \mathscr{V}_1' \times \mathscr{C}_2' \subseteq \mathscr{V}_1 \times \mathscr{C}_2$ in $\mathscr{G}_1 \times \mathscr{G}_2$,

$$N := \{\, x \in (\mathscr{V}_1 \times \mathscr{V}_2) \cup (\mathscr{C}_1 \times \mathscr{C}_2) \;\mid\; x \text{ is connected to } (v_1', c_2') \text{ by an edge in } \mathscr{G}_1 \times \mathscr{G}_2 \,\}.$$

We claim that $N = N'$.

Clearly, $N' \subseteq N$. Conversely, let $x = (x_1, x_2) \in N$. Assume $(x_1, x_2) \in \mathscr{V}_1 \times \mathscr{V}_2$ (the other case is analogous). Then (x_1, x_2) is connected to (v_1', c_2'), which implies $x_1 = v_1'$ (since $x_2 \neq c_2'$ being in $\mathscr{V}_2$). Thus $(x_1, x_2) \in \mathscr{V}_1' \times \mathscr{V}_2$, and by the definition of a subgraph, $x \in N'$. Hence $N = N'$.

Therefore, we also have

$$z = \bigoplus_{(v_1', c_2') \in J} h_Z(v_1', c_2'),$$

proving that z defines a Z-type stabilizer and hence $E_Z \in G$. A similar argument shows that $E_X \in G$. Therefore $E \in G$, completing the proof. □

Remark 3.28. *One can also show that*

$$d \leq \min(d_1, d_2, d_1^T, d_2^T).$$

Corollary 3.29. *Let $H_1 \in \mathrm{M}_{m_1 \times n_1}(\mathbb{Z}_2)$ and $H_2 \in \mathrm{M}_{m_2 \times n_2}(\mathbb{Z}_2)$ be the parity-check matrices of two linear codes C_1 and C_2 with parameters $[n_1, k_1, d_1]$ and $[n_2, k_2, d_2]$, respectively. Then the hypergraph product code $\mathrm{HGP}(H_1, H_2)$ has parameters*

$$[\![\, n_1 n_2 + m_1 m_2,\;\; k_1 k_2 + k_1^T k_2^T,\;\; \min(d_1, d_2, d_1^T, d_2^T) \,]\!].$$

Notes

- The presentation of the toric code and hypergraph product codes in these notes is largely based on a course by Debbie Leung and Michael Vasmer, available at *math.uwaterloo.ca* 2024.
- A proof of Remark 3.28 can be found in Tillich and Zemor 2009.

References for Lecture 7 & 8.

math.uwaterloo.ca (2024). https://www.math.uwaterloo.ca/~wcleung/qic890-w2024.html. [Accessed 16-06-2026].

Tillich, Jean-Pierre and Gilles Zemor (2009). "Quantum LDPC codes with positive rate and minimum distance proportional to $n^{\frac{1}{2}}$". In: *2009 IEEE International Symposium on Information Theory*, pp. 799–803. DOI: 10.1109/ISIT.2009.5205648.

4. Lecture 9 & 10: The Decoding Problem and the Surface Code

Recall that G_n is the n-qubit Pauli group. Fix a stabilizer group $G \subseteq G_n$. We then have

$$G \subseteq N(G) \subseteq G_n,$$

where $N(G)$ denotes the normalizer of the abelian group G. It is straightforward to check that $N(G)$ is a normal subgroup of G_n.

To define the syndrome of an element $E \in G_n$, we fix an independent generating set $\{g_1, g_2, \ldots, g_l\}$ of G. For each $j = 1, 2, \ldots, l$, set

$$s_j := \Omega(g_j) J \Omega(E)^T.$$

Then the bit string

$$s(E) := (s_1, s_2, \ldots, s_l) \in \mathbb{Z}_2^l$$

is called the *syndrome* of E. Hence we obtain a map

$$s : G_n \longrightarrow \mathbb{Z}_2^l.$$

One can check that s is a group homomorphism. Moreover, s is surjective, which follows from Exercise 3.11.

Now suppose $s(E) = s(F)$ for $E, F \in G_n$. Then $s(EF) = 0$, which means that $EF \in N(G)$.

Thus we obtain the following short exact sequence:

$$1 \longrightarrow N(G) \longrightarrow G_n \xrightarrow{s} \mathbb{Z}_2^l \longrightarrow 0.$$

Let us now explain the importance of the syndrome. Suppose $|\psi\rangle \in \mathcal{C}(G)$, and during computation an unknown error $E \in G_n$ is applied to the state, producing $E|\psi\rangle$. The goal of quantum error correction is to detect and correct this error.

For error detection, we proceed as follows. For each $j = 1, 2, \ldots, l$, consider the projectors

$$P_0^j := \frac{I + g_j}{2}, \qquad P_1^j := \frac{I - g_j}{2},$$

and perform a quantum measurement (in a non-destructive way) of these operators on the state $E|\psi\rangle$. It can be verified that if E commutes with g_j, the measurement outcome is always 0, while if E anticommutes with g_j, the outcome is always 1.

Hence, the syndrome vector

$$s = s(E) = (s_j)_{j=1}^l$$

can be regarded as the outcome of this measurement process. Once we obtain a syndrome, we try to infer the error E and apply a suitable recovery operator R so that

$$R^\dagger E |\psi\rangle = |\psi\rangle,$$

thereby returning to the original state $|\psi\rangle$, whenever this is possible. This is called the decoding process.

Given a syndrome $s = s(E)$, since s is a surjective map, we can always choose a correction (or recovery) operator $R \in G_n$ which produces the same syndrome as E, i.e. $s(R) = s(E)$. However, E and R need not be equal. In fact, from the previous discussion, E and R differ by an element of the normalizer $N(G)$, i.e. $RE \in N(G)$ or $R^\dagger E \in N(G)$. We say that the recovery operation is successful if $R^\dagger E \in G$. We record this in the following definition.

Definition 4.1. *Let G be a stabilizer group with stabilizer generators $\{g_1, g_2, \ldots, g_l\}$. A decoding strategy or decoder is defined by a map*

$$c : \mathbb{Z}_2^l \longrightarrow G_n, \quad s \longmapsto c(s),$$

which associates a correction operator $c(s) \in G_n$ to every syndrome $s \in \mathbb{Z}_2^l$.

(1) The decoder is valid for an error $E \in G_n$ if

$$s(c(s(E))) = s(E) \quad \iff \quad c(s(E))E \in N(G) \iff \quad c(s(E))^\dagger E \in N(G)..$$

(2) The decoder is successful for an error $E \in G_n$ if

$$c(s(E))^\dagger E \in G.$$

Recall the short exact sequence

$$1 \longrightarrow N(G) \longrightarrow G_n \xrightarrow{s} \mathbb{Z}_2^l \longrightarrow 0$$

A valid decoder is precisely a *section* of this exact sequence, i.e. a right inverse of the syndrome map s. In other words, we have

$$1 \longrightarrow N(G) \longrightarrow G_n \underset{c}{\overset{s}{\rightleftarrows}} \mathbb{Z}_2^l \longrightarrow 0$$

where the map c is a choice of section satisfying $s \circ c = I$.

Example 4.2. *Consider the three-qubit bit-flip code with stabilizer generators*

$$g_1 = \mathbf{ZZ}\mathrm{I}, \qquad g_2 = \mathrm{I}\mathbf{ZZ}.$$

Define the following decoding strategy

$$c : \mathbb{Z}_2^2 \longrightarrow G_3, \qquad c((0,0)) = I, \;\; c((0,1)) = \mathrm{II}\mathbf{X}, \;\; c((1,0)) = \mathbf{X}\mathrm{II}, \;\; c((1,1)) = \mathrm{I}\mathbf{X}\mathrm{I}.$$

It can be checked that this decoder is both valid and successful for any single-qubit X-error, i.e.,

$$E \in \{\mathbf{X}\mathrm{II}, \; \mathrm{I}\mathbf{X}\mathrm{I}, \; \mathrm{II}\mathbf{X}\}.$$

Note that for the error $E = \mathbf{XX}\mathrm{I}$, the above decoder is valid but not successful, since $\mathbf{XXX} \in N(G) \setminus G$.

In general, a mixed state $\rho \in \mathcal{B}(\mathcal{H})$ will pass through a quantum channel $\mathcal{E}$. We now describe how the recovery procedure works in this situation.

After applying the channel, we obtain the state $\mathcal{E}(\rho)$. We then measure the stabilizers on this state. Concretely, we measure the operators

$$P_s := \prod_{j=1}^{l} \frac{1}{2} \left(I + (-1)^{s_i} g_j\right), \qquad s \in \mathbb{Z}_2^l,$$

on $\mathcal{E}(\rho)$. The syndrome $s \in \mathbb{Z}_2^l$ occurs with probability

$$p(s) := \mathrm{Tr}(P_s \, \mathcal{E}(\rho)) \,,$$

and the corresponding post-measurement state is

$$\frac{1}{p(s)} \, P_s \mathcal{E}(\rho) P_s.$$

Let

$$c : \mathbb{Z}_2^l \longrightarrow G_n, \qquad s \longmapsto c(s)$$

be a decoding strategy. The "recovery" is then applied to the post-measurement state, yielding

$$\frac{1}{p(s)}\, c(s)^\dagger\, P_s \mathcal{E}(\rho) P_s\, c(s)$$

for the outcome s, which occurs with probability $p(s)$. Therefore, the final recovered state is

$$\mathcal{R} \circ \mathcal{E}(\rho) = \sum_{s \in \mathbb{Z}_2^l} c(s)^\dagger\, P_s\, \mathcal{E}(\rho)\, P_s\, c(s). \tag{4.1}$$

The recovery is *successful* if $\mathcal{R} \circ \mathcal{E}(\rho) \propto \rho$.

One may compare the above scenario with Theorem 2.20. To do this, assume that $\mathcal{E}$ is composed of noise operators $\{E_i\}_{i=1}^r$, such that the set $\{E_i^\dagger E_j\}_{i,j}$ is detectable. Let $s(i) := s(E_i)$ denote the syndrome of E_i. Then it is easy to see that $p(s) = 0$ unless $s = s(E_i)$ for some i.

Hence, we can re-index the expression 4.1 using i, which ranges from 1 to r. Thus, expression 4.1 becomes

$$\mathcal{R} \circ \mathcal{E}(\rho) = \sum_{i=1}^{r} c(s(i))^\dagger\, P_{s(i)}\, \mathcal{E}(\rho)\, P_{s(i)}\, c(s(i)). \tag{4.2}$$

Since $\{E_i^\dagger E_j\}_{i,j}$ are detectable, the recovery channel in Theorem 2.20 coincides with the above "recovery" for an appropriate choice of $c(i)$. In fact, one can take $c(s(i)) = E_i$, which is a unitary operator (appearing in the polar decomposition), since each E_i is a Pauli operator. Finally, note that $P_{s(i)}$ is precisely the projection P_i from the proof of Theorem 2.20.

Now let us consider a general Pauli noise channel (see Subsection 2.4)

$$\mathcal{E}_\pi(\rho) = \sum_{E \in G_n} \pi(E)\, E \rho E^\dagger,$$

where $\pi : G_n \to [0,1]$ is a probability distribution. Let c be a decoding strategy. We are interested in the *effective channel* $\mathcal{R} \circ \mathcal{E}_\pi$, which is given by

$$\mathcal{R} \circ \mathcal{E}_\pi(\rho) = \sum_{s \in \mathbb{Z}_2^l} c(s)^\dagger\, P_s\, \mathcal{E}_\pi(\rho)\, P_s\, c(s). \tag{4.3}$$

We now have the following theorem. Let $k = n - l$, i.e., the stabilizer code encodes k qubits. Hence there are 4^k logical Pauli operators up to phase, which we denote by l_i, $i = 0, 1, 2, \ldots, 4^k - 1$, with l_0 being the identity operator.

Theorem 4.3. *If c is a valid decoding strategy, then the effective channel $\mathcal{R} \circ \mathcal{E}_\pi$ takes the form*

$$\mathcal{R} \circ \mathcal{E}_\pi(\rho) = \sum_{i=0}^{4^k - 1} \pi_{\text{eff}}(l_i)\, l_i \rho l_i^\dagger, \tag{4.4}$$

where the effective distribution is given by

$$\pi_{\text{eff}}(l_i) = \sum_{\substack{E \in G_n \\ [c(s(E))^\dagger E] = [l_i]}} \pi(E). \tag{4.5}$$

Proof. We know that

$$\mathcal{R} \circ \mathcal{E}_\pi(\rho) = \sum_{s \in \mathbb{Z}_2^l} c(s)^\dagger\, P_s\, \mathcal{E}_\pi(\rho)\, P_s\, c(s).$$

Substituting $\mathcal{E}_\pi(\rho) = \sum_{E\in G_n} \pi(E)\, E\rho E^\dagger$, we observe that for fixed s: if $s(E) = s$, then $P_s E\rho E^\dagger P_s = E\rho E^\dagger$, and if $s(E) \neq s$, then $P_s E\rho E^\dagger P_s = 0$.

Thus,

$$\mathcal{R}\circ\mathcal{E}_\pi(\rho) = \sum_{s\in\mathbb{Z}_2^l} \sum_{\substack{E\in G_n \\ s(E)=s}} \pi(E)\, c(s)^\dagger\, E\,\rho\, E^\dagger\, c(s).$$

$$= \sum_{E\in G_n} \pi(E)\, c(s(E))^\dagger\, E\,\rho\, E^\dagger\, c(s(E)).$$

Since c is a valid decoder, for all E we have $c(s(E))^\dagger E \in N(G)$. Therefore, $[c(s(E))^\dagger E] = [l_i]$ for some $i = 0, 1, \ldots, 4^k - 1$. This implies that $c(s(E))^\dagger E$ and l_i differ only by a stabilizer element, and hence

$$c(s(E))^\dagger\, E\,\rho\, E^\dagger\, c(s(E)) = l_i \rho l_i^\dagger.$$

Reindexing the sum by i yields the desired form. □

Note that if the decoder c is successful for all E, then $\pi_{\text{eff}}(l_0) = 1$ and $\pi_{\text{eff}}(l_i) = 0$ for all other i. The quantity $\pi^c_{\text{eff}}(l_0)$ is called the *effective probability of correct decoding*. Similarly, the quantity

$$\sum_{i=1}^{4^k-1} \pi_{\text{eff}}(l_i)$$

is called the *effective probability of incorrect decoding*. Naturally, we would like to choose a decoder c such that $\pi^c_{\text{eff}}(l_0)$ is as large as possible.

Now we want to discuss two different kinds of decoder maps c. For each s, we need to define $c(s)$.

Fix, for each s, a *pure error* $E_s \in G_n$ such that $s(E_s) = s$. Then any $E \in G_n$ with $s(E) = s$ can be written in the form

$$E = E_s\, l_i\, g,$$

for some $i \in \{0, 1, \ldots, 4^k - 1\}$ and $g \in G$.

We now define a decoder, called the *suboptimal decoder*, by

$$c(s) = \arg\max_{i,g}\, \pi(E_s l_i g).$$

Let us check that it is a valid decoder. $c(s(E))^\dagger E = g l_j E_s E_s l_i g' = g l_j l_i g' \in N(G)$, where $s = s(E)$.

Next we define another decoder which is "better" than the above suboptimal decoder. For this, for each s, we first define

$$\widetilde{\pi}(E_s l_i) = \sum_{g\in G} \pi(E_s l_i g).$$

We then define a decoder, called the *maximum likelihood decoder*, by

$$c_{\text{ml}}(s) = \arg\max_{i}\, \widetilde{\pi}(E_s l_i).$$

Now we want to show that the above maximum likelihood decoder performs better than any other valid decoder. Let c be any valid decoder and let π^c_{eff} denote the effective distribution corresponding to c. Similarly, let $\pi^{c_{\text{ML}}}_{\text{eff}}$ denote the effective distribution corresponding to c_{ml}. We then have the following theorem.

Theorem 4.4. *For any valid decoder c, we have*

$$\pi^{c_{\text{ml}}}_{\text{eff}}(l_0) \;\geq\; \pi^c_{\text{eff}}(l_0).$$

Proof. For any valid decoder c, the effective probability of the identity is

$$\pi_{\text{eff}}^{c}(l_0) = \sum_{\substack{E\in G_n \\ [c(s(E))^\dagger E]=[l_0]}} \pi(E). \tag{4.6}$$

Suppose the decoder is of the form

$$c(s) = E_s l_c \tilde{g},$$

for some logical operator l_c and some $\tilde{g} \in G$. Then it follows that

$$\pi_{\text{eff}}^{c}(l_0) = \sum_{s} \sum_{g\in G} \pi(E_s l_c g).$$

On the other hand, for the maximum likelihood (ML) decoder we have

$$\pi_{\text{eff}}^{c_{\text{ml}}}(l_0) = \sum_{s} \sum_{g\in G} \pi(E_s l_{c_{\text{ML}}} g).$$

By the definition of the ML decoder,

$$\sum_{g\in G} \pi(E_s l_{c_{\text{ml}}} g) \;\geq\; \sum_{g\in G} \pi(E_s l_c g),$$

for each syndrome s.

Summing over all s gives

$$\pi_{\text{eff}}^{c_{\text{ml}}}(l_0) \;\geq\; \pi_{\text{eff}}^{c}(l_0),$$

which proves the claim. □

Exercise 4.5. ★ *Consider the one-qubit bit-flip channel defined by*

$$\mathcal{E}(\rho) = p\,\mathbf{X}\rho\mathbf{X} + (1-p)\rho,$$

where $p \in [0,1]$. On three qubits, consider the independent channel

$$\mathcal{E} = \mathcal{E}_1 \otimes \mathcal{E}_2 \otimes \mathcal{E}_3.$$

Now consider the three-qubit bit-flip code from Exercise 4.2 together with the decoding strategy c defined therein.

(1) Compute $\pi_{\text{eff}}^{c}(l_0)$, the effective probability of correct decoding.
(2) Determine the set of p for which

$$\pi_{\text{eff}}^{c}(l_0) > 1 - p.$$

(3) Compare this result with the threshold computation in the classical repetition code (Example 2.8).

4.1. **Error Correction with the Toric Code.** Let us consider the $L \times L$ toric code. For the error channel, we take the independent channel

$$\mathcal{F} = \mathcal{E}^{\otimes 2L^2},$$

where $\mathcal{E}$ is the one-qubit phase-flip channel defined by

$$\mathcal{E}(\rho) = p\,\mathbf{Z}\rho\mathbf{Z} + (1-p)\rho, \qquad p \in [0,1].$$

We also define the *subchannel* $\widetilde{\mathcal{F}}$ of $\mathcal{F}$ obtained by restricting to those Kraus operators of $\mathcal{F}$ whose weight is strictly less than $L/2$.

Let c denote the suboptimal decoder for the toric code. We claim that for the channel $\widetilde{\mathcal{F}}$, we have

$$\pi_{\text{eff}}^{c}(l_0) = 1,$$

which means that c perfectly corrects all errors arising from $\widetilde{\mathcal{F}}$. It suffices to show that for every error E appearing in the Kraus decomposition of $\widetilde{\mathcal{F}}$, we have

$$c(s(E))^{\dagger}E \in G.$$

Since c is a valid decoder, we already know that

$$c(s(E))^{\dagger}E \in N(G).$$

Suppose, for the sake of contradiction, that

$$c(s(E))^{\dagger}E \in N(G) \setminus G,$$

i.e., it represents a nontrivial logical operator. Any nontrivial logical operator of the toric code has weight at least L, so

$$\text{wt}\big(c(s(E))^{\dagger}E\big) \geq L.$$

Since $\text{wt}(E) < L/2$ by assumption, this forces

$$\text{wt}\big(c(s(E))\big) > L/2.$$

Consequently,

$$\pi(c(s(E))) < p^{L/2}, \qquad \pi(E) > p^{L/2}.$$

This contradicts the definition of the suboptimal decoder. Hence, no such E can exist, and we must have $c(s(E))^{\dagger}E \in G$ for every E in the Kraus decomposition of $\widetilde{\mathcal{F}}$. Therefore, $\pi_{\text{eff}}^{c}(l_0) = 1$.

Now let us consider the full channel $\mathcal{F}$ with the same decoding strategy c. We are interested in the quantity

$$\sum_{i=1}^{4^k-1} \pi_{\text{eff}}(l_i),$$

which is the *effective probability of incorrect decoding.*

By definition,

$$\begin{aligned}\sum_{i=1}^{4^k-1} \pi_{\text{eff}}(l_i) &= \sum_{i=1}^{4^k-1} \sum_{\substack{E\in G_n \\ [c(s(E))^{\dagger}E]=[l_i]}} \pi(E) \\ &\leq \sum_{l\geq L} \sum_{\substack{E\in G_n \\ \text{wt}(c(s(E))^{\dagger}E)=l}} \pi(E). \qquad (4.7)\end{aligned}$$

The inequality holds because a logical error must correspond to a representative $c(s(E))^{\dagger}E$ of weight at least L, the code distance of the toric code.

Bounding the probability. For a fixed l, the contribution can be overestimated by counting all possible error loops of length l:

$$\sum_{i=1}^{4^k-1} \pi_{\text{eff}}(l_i) \;\leq\; \sum_{l\geq L} \big(\#\text{loops of length } l\big) \cdot 2^l p^{l/2}.$$

Here the factor 2^l accounts for the number of ways E can occur on a loop of length l, the factor $p^{l/2}$ appears because there must be at least $l/2$ errors on the loop to produce a logical error.

Next, we bound the number of loops of length l by an overestimate:

$$\#\text{loops of length } l \;\leq\; l^2 \cdot 4 \cdot 3^{l-1},$$

where l^2 accounts for the choice of a starting edge on the $L \times L$ lattice, the factor 4 accounts for the choice of an initial direction, 3^{l-1} counts the number of possible continuations at each subsequent step (avoiding backtracking).

Thus,

$$\sum_{i=1}^{4^k-1} \pi_{\text{eff}}(l_i) \;\leq\; \sum_{l\geq L} l^2 \cdot 4 \cdot 3^{l-1} \cdot 2^l p^{l/2}.$$

This can be simplified as

$$\sum_{i=1}^{4^k-1} \pi_{\text{eff}}(l_i) \;\leq\; \sum_{l\geq L} \frac{4L^2}{3}\, 6^l p^{l/2} = \frac{4L^2}{3} \sum_{l\geq L} (36p)^{l/2}.$$

Let

$$p_0 := \frac{1}{36}.$$

Then

$$\sum_{i=1}^{4^k-1} \pi_{\text{eff}}(l_i) \;\leq\; \frac{4L^2}{3} \left(\frac{p}{p_0}\right)^{L/2} \frac{1}{1-\sqrt{p/p_0}}.$$

This bound shows that as long as $p < p_0$, the effective probability of incorrect decoding decays exponentially in L. In particular, for sufficiently large lattice size L, the probability of successful decoding

$$\pi^c_{\text{eff}}(l_0) = 1 - \sum_{i=1}^{4^k-1} \pi_{\text{eff}}(l_i)$$

can be made arbitrarily close to 1.

Notes

- The mathematical formulation of the decoding problem presented here is based on the PhD thesis of Margret Heinze; see Heinze 2023.
- The discussion of toric code decoding presented here is again based on *math.uwaterloo.ca* 2024. Readers interested in a more comprehensive treatment of decoding algorithms for such codes are referred to iOlius et al. 2024.

References for Lecture 9 & 10.

Heinze, Margret (2023). "Quantum Fault-Tolerance with Continuous Variable Systems". PhD thesis.

iOlius, Antonio deMarti et al. (Oct. 2024). "Decoding algorithms for surface codes". In: *Quantum* 8, p. 1498. ISSN: 2521-327X. DOI: 10.22331/q-2024-10-10-1498. URL: https://doi.org/10.22331/q-2024-10-10-1498.

math.uwaterloo.ca (2024). https://www.math.uwaterloo.ca/~wcleung/qic890-w2024.html. [Accessed 16-06-2026].

5. Lecture 11 & 12: Quantum Gates on Codes

In this section, we aim to understand how quantum gates act on quantum codes. Recall that a *quantum gate* is represented by a unitary operator acting on the corresponding quantum system. Let $\mathcal{H}$ be the Hilbert space of n qubits, so that $\mathcal{H} = \mathbb{C}^{2^n}$. Then a quantum gate acting on $\mathcal{H}$ is a unitary matrix

$$U \in \mathcal{U}(2^n),$$

where $\mathcal{U}(2^n)$ denotes the group of all $2^n \times 2^n$ unitary matrices.

Exercise 5.1. ★ *Show that any unitary matrix $U \in \mathcal{U}(2^n)$ can be written as*

$$U = c\widetilde{U},$$

where $\widetilde{U}$ is a unitary matrix with determinant 1. Hence, conclude that up to a global phase, any quantum gate can be represented by a unitary matrix with determinant 1. (The set of such matrices is denoted by $\mathcal{SU}(2^n)$.)

5.1. **Examples of Unitary Gates.** Let us now give some examples of unitary gates. For a single qubit, the Pauli matrices $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are, of course, unitary operators.

The Hadamard transform is given by

$$\mathbf{H} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix},$$

which is also a single-qubit unitary operator. Note that

$$\mathbf{HXH}^\dagger = \mathbf{Z}, \qquad \mathbf{HZH}^\dagger = \mathbf{X}, \qquad \mathbf{H}^\dagger = \mathbf{H}. \tag{5.1}$$

Two other important examples of single-qubit unitary gates are the *phase gate* $\mathbf{S}$ and the *T-gate*, defined by

$$\mathbf{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \qquad \mathbf{T} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.$$

It is straightforward to check that

$$\mathbf{SXS}^\dagger = \mathbf{Y}, \qquad \mathbf{SZS}^\dagger = \mathbf{Z}, \qquad \mathbf{TXT}^\dagger = \tfrac{1}{\sqrt{2}}(\mathbf{X}+\mathbf{Y}), \qquad \mathbf{TZT}^\dagger = \tfrac{1}{\sqrt{2}}(\mathbf{Y}-\mathbf{X}). \tag{5.2}$$

For any j with $1 \le j \le n$, we can define n-qubit operators $\mathbf{S}_j$ and $\mathbf{H}_j$ acting on the j-th qubit in the natural way (and acting trivially on the other qubits).

Now consider examples of two-qubit gates. The *controlled-NOT* (CNOT) gate is defined by the matrix

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

One can easily verify that

$$\mathbf{CNOT}(\mathbf{X}\otimes \mathrm{I})\mathbf{CNOT}^\dagger = \mathbf{X}\otimes\mathbf{X}, \qquad \mathbf{CNOT}(\mathrm{I}\otimes\mathbf{X})\mathbf{CNOT}^\dagger = \mathrm{I}\otimes\mathbf{X}, \tag{5.3}$$

and

$$\mathbf{CNOT}(\mathbf{Z}\otimes \mathrm{I})\mathbf{CNOT}^\dagger = \mathbf{Z}\otimes\mathrm{I}, \qquad \mathbf{CNOT}(\mathrm{I}\otimes\mathbf{Z})\mathbf{CNOT}^\dagger = \mathbf{Z}\otimes\mathbf{Z}. \tag{5.4}$$

For any j, k with $1 \le j, k \le n$, we can define an n-qubit operator $\mathbf{CNOT}_{j,k}$, which acts as the standard **CNOT** gate on the j-th (control) and k-th (target) qubits, and as the identity on all other qubits.

Another important two-qubit gate related to the **CNOT** gate is the *swap gate*, denoted by **SWAP**. It is defined as

$$\mathbf{SWAP}_{j,k} := \mathbf{CNOT}_{j,k}\,\mathbf{CNOT}_{k,j}\,\mathbf{CNOT}_{j,k}.$$

As the name suggests, the swap gate $\mathbf{SWAP}_{j,k}$ exchanges the states of the j-th and k-th qubits.

Exercise 5.2. ★ *Verify that the 2-qubit* **SWAP** *exchanges the two qubits on which it acts. In particular, show that*

$$\mathbf{SWAP}\,(\mathbf{X}\otimes \mathrm{I})\,\mathbf{SWAP}^\dagger = \mathrm{I}\otimes\mathbf{X}, \qquad \mathbf{SWAP}\,(\mathrm{I}\otimes\mathbf{X})\,\mathbf{SWAP}^\dagger = \mathbf{X}\otimes\mathrm{I},$$

and

$$\mathbf{SWAP}\,(\mathbf{Z}\otimes \mathrm{I})\,\mathbf{SWAP}^\dagger = \mathrm{I}\otimes\mathbf{Z}, \qquad \mathbf{SWAP}\,(\mathrm{I}\otimes\mathbf{Z})\,\mathbf{SWAP}^\dagger = \mathbf{Z}\otimes\mathrm{I}.$$

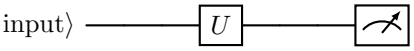
Moreover, check that

$$\mathbf{SWAP}\,(\mathbf{X}\otimes \mathbf{Z})\,\mathbf{SWAP}^\dagger = \mathbf{Z}\otimes\mathbf{X}, \qquad \mathbf{SWAP}\,(\mathbf{Z}\otimes\mathbf{X})\,\mathbf{SWAP}^\dagger = \mathbf{X}\otimes\mathbf{Z},$$

and finally that

$$\mathbf{SWAP}\,(\mathbf{X}\otimes \mathbf{X})\,\mathbf{SWAP}^\dagger = \mathbf{X}\otimes\mathbf{X}, \qquad \mathbf{SWAP}\,(\mathbf{Z}\otimes\mathbf{Z})\,\mathbf{SWAP}^\dagger = \mathbf{Z}\otimes\mathbf{Z}.$$

5.2. **Quantum Circuits and Universal Gate Sets.** As discussed earlier, a quantum computation is typically represented by a *quantum circuit*, which can be depicted as



where U is a unitary gate acting on the system.

Example: Quantum Teleportation. Let us now describe a concrete example of a quantum circuit — the *quantum teleportation* protocol. Suppose Alice wishes to send an unknown quantum state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ to Bob, but they are only allowed to communicate through a classical channel. Quantum teleportation demonstrates that this is possible if they share a Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ in advance.

Let

$$|\psi_0\rangle = |\psi\rangle \otimes \left|\Phi^+\right\rangle$$

be the initial joint state that Alice possesses (she holds $|\psi\rangle$ and the first qubit of $|\Phi^+\rangle$). Alice first applies a **CNOT** gate on her two qubits, using $|\psi\rangle$ as the control and her share of $|\Phi^+\rangle$ as the target. After this step, the state becomes

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}\big(\alpha\,|000\rangle + \alpha\,|011\rangle + \beta\,|110\rangle + \beta\,|101\rangle\big).$$

Next, she applies a Hadamard gate **H** on the first qubit, giving

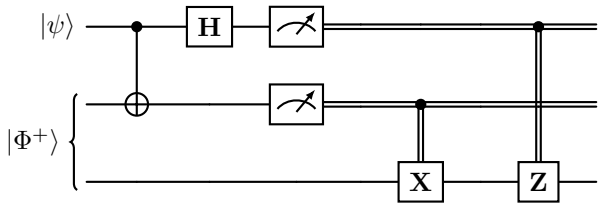
$$|\psi_2\rangle = \frac{1}{2}\Big(|00\rangle\,(\alpha\,|0\rangle + \beta\,|1\rangle) + |01\rangle\,(\alpha\,|1\rangle + \beta\,|0\rangle) + |10\rangle\,(\alpha\,|0\rangle - \beta\,|1\rangle) + |11\rangle\,(\alpha\,|1\rangle - \beta\,|0\rangle)\Big).$$

Alice then measures her two qubits in the computational basis, obtaining two classical bits 00, 01, 10, or 11. She transmits these bits to Bob over a classical channel. Upon receiving the bits, Bob applies one of the following correction operators to his qubit:

Alice's measurement outcome	Bob's correction
00	I
01	**X**
10	**Z**
11	**ZX**

After the correction, Bob's qubit is exactly $|\psi\rangle$.

The complete quantum teleportation circuit is shown below:



Exercise 5.3. ★ *Write down the detailed step-by-step computation of the quantum teleportation protocol, verifying that Bob's final qubit state is indeed $|\psi\rangle$ for all measurement outcomes.*

5.3. **Quantum Circuits and Universal Gate Sets.** We now formally define a *quantum circuit*.

Definition 5.4 (Quantum Circuit)**.** *A* quantum circuit *C on n qubits is a sequence of unitary operators*

$$U_1, U_2, \ldots, U_L$$

acting on the n-qubit Hilbert space $\mathcal{H} = \mathbb{C}^{2^n}$, such that each U_i acts nontrivially on at most two qubits. The overall action of the circuit is given by

$$C = U_L U_{L-1} \cdots U_1.$$

The nontrivial part of each U_i is denoted by $\widetilde{U}_i$, which is either a one-qubit or a two-qubit unitary operator.

Example. *In the quantum teleportation circuit, we have for instance*

$$U_1 = \mathbf{CNOT} \otimes \mathrm{I}, \qquad U_2 = \mathbf{H} \otimes \mathrm{I} \otimes \mathrm{I}.$$

Note that in the definition of a quantum circuit, we consider only the unitary operations applied before *the measurement process.*

Definition 5.5 (Universal Gate Set)**.** *(see also* URL of ICTP SAIFR *2022) A finite set S of one-qubit and two-qubit unitary gates is called a* universal gate set *if for every $n \in \mathbb{N}$, every unitary $U \in \mathcal{U}(2^n)$, and every $\epsilon > 0$, there exists a quantum circuit*

$$C = U_L U_{L-1} \cdots U_1, \qquad \text{with } \widetilde{U}_i \in S,$$

such that

$$\|U - C\| < \epsilon.$$

The following theorem gives a very important example of a universal gate set.

Theorem 5.6. *(cf. Boykin et al. 2000) The set $\{\mathbf{H}, \mathbf{S}, \mathbf{CNOT}, \mathbf{T}\}$ forms a universal gate set.*

5.4. **Clifford Gates and the Gottesman–Knill Theorem.** In the universal gate set $S = \{\mathbf{H}, \mathbf{S}, \mathbf{CNOT}, \mathbf{T}\}$, the subset $\{\mathbf{H}, \mathbf{S}, \mathbf{CNOT}\}$ plays a particularly important role. We will now discuss why these gates are special.

Lemma 5.7. *Let $G = \langle g_1, \ldots, g_l \rangle$ be a subgroup of the n-qubit Pauli group G_n, generated by independent elements $g_1, g_2, \ldots, g_l$. Suppose U is a unitary operator in $\mathcal{U}(2^n)$. Then the subspace*

$$\{U|\psi\rangle \mid |\psi\rangle \in C(G)\}$$

is stabilized by the operators $Ug_1U^\dagger, Ug_2U^\dagger, \ldots, Ug_lU^\dagger$.

Proof. The proof is straightforward. Let $|\psi\rangle \in \mathcal{C}(G)$, so that $g_i |\psi\rangle = |\psi\rangle$ for all i. Then for each generator g_i, we have

$$U|\psi\rangle = Ug_i|\psi\rangle = (Ug_iU^\dagger)U|\psi\rangle .$$

Hence, $U|\psi\rangle$ is stabilized by $Ug_iU^\dagger$, proving the claim. □

However, in general, it is not guaranteed that $Ug_iU^\dagger$ belongs to the Pauli group G_n. Therefore, the image subspace

$$\{U|\psi\rangle \mid |\psi\rangle \in \mathcal{C}(G)\}$$

need not be a stabilizer subspace. This observation motivates the following definition.

Definition 5.8. *The set of all unitaries $U \in \mathcal{U}(2^n)$ such that*

$$UgU^\dagger \in G_n \quad \textit{for all } g \in G_n$$

is called the Clifford group. *It is denoted by* CL_n.

It is easy to verify that the Clifford group CL_n is indeed a group under composition. The circle group $U(1)$ sits inside $\mathrm{CL_n}$ as a normal subgroup. We define $\widetilde{\mathrm{CL}}_n$ to be the quotient group $\mathrm{CL_n/U(1)}$.

Of course, all the elements of G_n are also in $\mathrm{CL_n}$. Also it is clear from the relations 5.1,5.2,5.3,5.4 that $\mathbf{H}_j$, $\mathbf{S}_j$, $\mathbf{CNOT}_{j,k}$ for j,k, with $1 \leq j,k \leq n$, are in the Clifford group $\mathrm{CL_n}$. Also note that $\mathbf{T}$ is not in $\mathrm{CL_n}$. In fact the following strong statement is true.

Then we have the following theorem.

Theorem 5.9. *(cf. Nielsen and Chuang 2010, Theorem 10.6) The set*

$$\{\mathbf{H}_j, \mathbf{S}_j, \mathbf{CNOT}_{j,k} \mid 1 \leq j,k \leq n,\}$$

generate $\widetilde{\mathrm{CL}}_n$.

Exercise 5.10. ★ *Prove the above theorem for $n = 1$.*

In the following we will see how a quantum circuit with Clifford group operations works. Suppose we have a initial state $|0\rangle^{\otimes n}$. Then this state is the unique state (up to a global phase) stabilized by the operators $\{\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_n\}$. Now we want to understand the evolution of the initial state under Clifford group operations. Suppose at some point in the circuit we have a state $|\psi\rangle$ with unique stabilizers $\{g_1, g_2, \cdots, g_n\}$, and we have an updated state $U|\psi\rangle$, where U is some Clifford unitary (gate), then it is easy to see that the stabilizers for the updated state $U|\psi\rangle$ will be $\{Ug_1U^\dagger, Ug_2U^\dagger, \cdots, Ug_nU^\dagger\}$. So the point is that we can track the state just by tracking the stabilizers. Now if we want to understand the measurement of a state $|\psi\rangle$ with unique stabilizers $\{g_1, g_2, \cdots, g_n\}$, for any measurement $g \in G_n$. Note that g is Hermitian, and hence $g^2 = \mathrm{I}$. Note that g has eigenvalues ± 1. The projections corresponding to the eigenvalues are $\frac{(\mathrm{I}+g)}{2}$, $\frac{(\mathrm{I}-g)}{2}$, respectively. is Now we have two cases:

Case 1. g commutes with g_i for all i, $1 \leq i \leq n$. We want to compute $\langle\psi| g |\psi\rangle$. In this case, first note that $g|\psi\rangle$ is stabilised by g_i for all i. This means that $g|\psi\rangle$ is a multiple of $|\psi\rangle$. But since $g^2 = \mathrm{I}$, $g|\psi\rangle = |\psi\rangle$ or $g|\psi\rangle = -|\psi\rangle$, whence g or $-g$ is in the group $\langle g_1, g_2, \cdots, g_n\rangle$. generated by $\{g_1, g_2, \cdots, g_n\}$. Then $\langle\psi| g |\psi\rangle = 1$ if $g \in \langle g_1, g_2, \cdots, g_n\rangle$, $\langle\psi| g |\psi\rangle = -1$ if $-g \in \langle g_1, g_2, \cdots, g_n\rangle$. In both cases, since the probability of occurring 1 or -1 is one, the final state would be $|\psi\rangle$, and hence we do not update the generators $\{g_1, g_2, \cdots, g_n\}$.

Case 2. g anti-commutes with one or more g_i. WLOG we may assume that g anti-commutes with g_1 and commutes with $g_2, \ldots, g_n$. This we can do since if g anti-commutes with one of

the other elements, say g_2, then g commutes with $g_1 g_2$. Then we can replace the generator g_2 by $g_1 g_2$ and update the list $\{g_1, g_2, \cdots, g_n\}$.

Using exactly the same calculation as in Equation 3.5 we conclude that $\langle\psi| g |\psi\rangle = 0$, which gives that the probability of getting 1 from the outcome is $\frac{1}{2}$ and probability of getting -1 from the outcome is $\frac{1}{2}$. If we get 1, then the updated state is $\frac{(\mathrm{I}+g)|\psi\rangle}{\sqrt{2}}$ with the updated generators $\{g, g_2, \cdots, g_n\}$. If we get 1, then the state is $\frac{(\mathrm{I}-g)|\psi\rangle}{\sqrt{2}}$ with the generators $\{-g, g_2, \cdots, g_n\}$.

Remark 5.11. The above theorem and the above discussion regarding a quantum circuit using Clifford group operations give us the following result which is known as the Gottesman–Knill theorem. The above discussions implicitly prove the theorem.

Suppose we perform a quantum computation which involves the following:

(1) state preparations in the computational basis (or any stabilizer state);
(2) a quantum circuit involving the Clifford gates (the theorem above shows that the gates only consist of Hadamard gates, phase gates, CNOT gates, Pauli gates);
(3) measurements of observables in the Pauli group.

Such a computation can be simulated efficiently on a classical computer.

Of course we do not expect to simulate every quantum computation in a classical computer (otherwise we will not be studying quantum computation at all :)), but Gottesman–Knill theorem shows that there is a class of quantum computation for which we can really use a classical computer.

5.5. **Unitary Gates on Quantum Error-Correcting Codes.** Let $\mathcal{C}$ be a quantum code subspace of a Hilbert space $\mathcal{H}$. Suppose we wish to apply a logical gate U on the encoded information in $\mathcal{C}$. Our goal is to determine a physical unitary operator $\overline{U}$ acting on the full Hilbert space $\mathcal{H}$ such that the action of $\overline{U}$ on code states correctly implements the logical transformation U on $\mathcal{C}$. This motivates the following definition.

Definition 5.12. *Let $\mathcal{C} \subseteq \mathcal{H}$ be a quantum code, and let*

$$U : \mathcal{C} \to \mathcal{C}$$

be a unitary operator acting on the logical code space. A unitary operator $\overline{U}$ acting on the full Hilbert space $\mathcal{H}$ is said to implement *U on the code $\mathcal{C}$ if the following diagram commutes:*

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{\quad U \quad} & \mathcal{C} \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle i} \\ \mathcal{H} & \xrightarrow{\quad \overline{U} \quad} & \mathcal{H} \end{array}$$

where $i : \mathcal{C} \hookrightarrow \mathcal{H}$ denotes the natural inclusion map.

Equivalently, $\overline{U}$ implements U on the code if for all $|\psi\rangle \in \mathcal{C}$,

$$\overline{U} |\psi\rangle = U |\psi\rangle .$$

Exercise 5.13. ★ *Show that there always exists a unitary $\overline{U}$ on $\mathcal{H}$ that implements U on $\mathcal{C}$ in the sense of the above definition.*

Recall the 3-qubit repetition code (also called the bit-flip code), which encodes one logical qubit into three physical qubits. The logical basis states are

$$|0_L\rangle = |000\rangle , \qquad |1_L\rangle = |111\rangle ,$$

and the stabilizer group is generated by

$$G = \langle \mathbf{ZZI},\ \mathrm{I}\mathbf{ZZ} \rangle.$$

Logical Pauli Operators. As we have seen earlier, we may choose the logical Pauli operators as

$$\overline{\mathbf{X}} = \mathbf{XXX}, \qquad \overline{\mathbf{Z}} = \mathbf{ZZZ}.$$

Indeed,

$$\overline{X}\,|0_L\rangle = |1_L\rangle\,, \qquad \overline{X}\,|1_L\rangle = |0_L\rangle\,,$$

and

$$\overline{Z}\,|0_L\rangle = |0_L\rangle\,, \qquad \overline{Z}\,|1_L\rangle = -\,|1_L\rangle\,,$$

as required.
Logical Phase Gate. We can also construct a logical phase gate $\overline{\mathbf{S}}$ that implements the single-qubit phase gate $\mathbf{S}$ on the encoded qubit. It is straightforward to check that

$$\overline{\mathbf{S}} = \mathbf{S}_1^\dagger \mathbf{S}_2 \mathbf{S}_3$$

acts as the desired logical phase gate on the code space.

Exercise 5.14. ★ *Construct an operator $\overline{\mathbf{H}}$ acting on three qubits that correctly implements the logical Hadamard gate on the repetition code.*

One possible candidate is given by

$$\overline{\mathbf{H}} := U_{\text{cat}}^\dagger\,(\mathbf{H}\otimes \mathrm{I}\otimes \mathbf{H})\,U_{\text{cat}}, \qquad \textit{where} \qquad \boldsymbol{U}_{\text{cat}} := \mathbf{CNOT}_{1,2}\,\mathbf{CNOT}_{1,3}.$$

Check that this operator satisfies

$$\overline{\mathbf{H}}\,|0_L\rangle = |+_L\rangle\,, \qquad \overline{\mathbf{H}}\,|1_L\rangle = |-_L\rangle\,.$$

Since we have already seen that the existence of an implementing operator $\overline{U}$ is not an issue, the real difficulty lies in *finding* such an operator. It is important to observe that in the example of the 3-qubit repetition code, the logical gates $\overline{\mathbf{X}}, \overline{\mathbf{Z}}, \overline{\mathbf{S}}$ we constructed are tensor products of single-qubit unitary operators, whereas the logical Hadamard gate $\overline{\mathbf{H}}$ required the use of two-qubit operations.

There is a clear advantage to using the former type of logical gates. Suppose we have a quantum circuit encoded using the repetition code, and a single-qubit X error occurs on one physical qubit. If we apply a logical gate that is a tensor product of single-qubit operators, then this error remains confined to that same physical qubit. However, this is generally not true for gates such as $\overline{\mathbf{H}}$ that involve two-qubit operations: the single-qubit error can propagate into a two-qubit error, which the code is no longer able to correct.

This motivates the following definition of *transversal gates*. For simplicity, we present the definition for the case of $k = 1$ logical qubit, although it easily generalizes to $k > 1$.

Definition 5.15. *Let $\mathcal{C}$ be a quantum code encoding $k = 1$ logical qubit. Let $U \in \{\mathbf{H}, \mathbf{S}, \mathbf{T}\}$ be a single-qubit logical gate, and let $\overline{U}$ be its physical implementation on $\mathcal{H}$. We say that $\overline{U}$ is* transversal *if it can be written as*

$$\overline{U} = U_1 \otimes U_2 \otimes \cdots \otimes U_n,$$

where each U_i is a single-qubit unitary operator.

We now extend the notion of transversality to the logical **CNOT** gate.

Definition 5.16. *Let $\mathcal{C}$ be a quantum code encoding $k = 1$ logical qubit, and consider two encoded blocks of the code $\mathcal{C}$. $\overline{\mathbf{CNOT}}$ acting between the two code blocks is said to be* transversal *if it can be written as*

$$\overline{\mathbf{CNOT}} = U_1 \otimes U_2 \otimes \cdots \otimes U_n,$$

where each U_i acts on at most one qubit from each code block.

It is clear that the logical operators $\overline{\mathbf{X}}$ and $\overline{\mathbf{Z}}$ can always be chosen to be transversal, since they can be constructed using the binary symplectic representation as we have seen earlier.

Exercise 5.17. ★ *Show that for any CSS code, the logical* **CNOT** *gate can be chosen to be transversal.*

Recall the Steane code from Example 3.6. A set of stabilizer generators is given by

$$g_1 = \mathrm{III}\mathbf{XXXX}, \quad g_2 = \mathrm{I}\mathbf{XX}\mathrm{II}\mathbf{XX}, \quad g_3 = \mathbf{X}\mathrm{I}\mathbf{X}\mathrm{I}\mathbf{X}\mathrm{I}\mathbf{X},$$

$$g_4 = \mathrm{III}\mathbf{ZZZZ}, \quad g_5 = \mathrm{I}\mathbf{ZZ}\mathrm{II}\mathbf{ZZ}, \quad g_6 = \mathbf{Z}\mathrm{I}\mathbf{Z}\mathrm{I}\mathbf{Z}\mathrm{I}\mathbf{Z}.$$

We can also visualize the stabilizer generators of the Steane code using a tessellation of a triangle. See Figure 10. The qubits are placed on the vertices, and an X-type stabilizer and a Z-type stabilizer are assigned to each face.

Exercise 5.18. ★ *Show that the distance of the Steane code is* 3*. Determine explicit expressions for the logical operators $\overline{\mathbf{X}}$ and $\overline{\mathbf{Z}}$.*

Exercise 5.19. ★ *Show that if $U \in \{\mathbf{H}, \mathbf{S}, \mathbf{CNOT}\}$, then $\overline{U}$ for the Steane code can be implemented transversally.*

Since all elements of the Clifford group (up to global phases) are generated by the set $\{\mathbf{H}, \mathbf{S}, \mathbf{CNOT}\}$, we conclude that every logical Clifford operation can be implemented transversally using the Steane code.

A natural question now arises: can the logical **T** gate also be implemented transversally for the Steane code? If this were possible, then the Steane code would support universal quantum computation using only transversal gates, since the set $\{\mathbf{H}, \mathbf{S}, \mathbf{CNOT}, \mathbf{T}\}$ is a universal gate set.

Unfortunately, this is not the case. A no-go theorem due to Eastin and Knill states that there is no quantum error-correcting code (capable of correcting at least one error) that admits transversal implementation of all gates from the universal set $\{\mathbf{H}, \mathbf{S}, \mathbf{CNOT}, \mathbf{T}\}$.

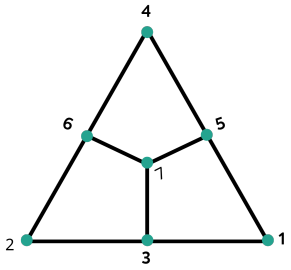


FIGURE 10

Next, we construct a quantum code for which the **T** gate can be implemented transversally. This code is known as the *15-qubit Reed–Muller code*, and it can be viewed as an example of a 3D color code.

To define this code, consider a tetrahedron and a suitable tessellation of it into smaller cells (see Figure 11: We place qubits on the vertices of the tessellated tetrahedron, assign X-type stabilizers to the 3-dimensional cells, and Z-type stabilizers to the 2-dimensional faces.

Exercise 5.20. ★ *In this exercise, you will verify basic parameters of the 15-qubit Reed–Muller code.*

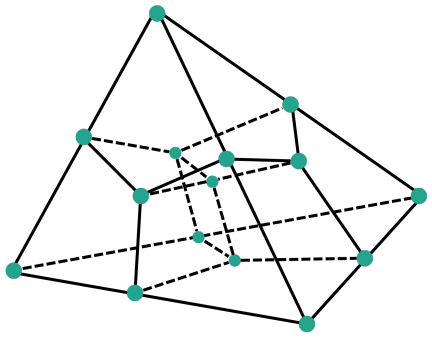


FIGURE 11

(1) *Show that there are* 14 *independent stabilizer generators:* X*-type stabilizers supported on the cells, and* Z*-type stabilizers supported on the faces. Conclude that the code encodes* $k = 1$ *logical qubit.*
(2) *Show that the logical operators can be chosen as*

$$\overline{\mathbf{X}} = \mathbf{X}^{\otimes 15}, \qquad \overline{\mathbf{Z}} = \mathbf{Z}^{\otimes 15}.$$

The following exercise shows that the $\mathbf{T}$ gate can be implemented transversally in the 15-qubit Reed–Muller code.

Exercise 5.21. ★

(1) *Show that the logical state* $|0_L\rangle$ *is a superposition of computational basis states of Hamming weight* 0 *or* 8*, and that the logical state* $|1_L\rangle$ *is a superposition of computational basis states of Hamming weight* 7 *or* 15*.*
(2) *Define the transversal operator*

$$\overline{\mathbf{T}} := (\mathbf{T}^\dagger)^{\otimes 15}.$$

Show that

$$\overline{\mathbf{T}}\,|0_L\rangle = |0_L\rangle\,, \qquad \overline{\mathbf{T}}\,|1_L\rangle = e^{i\pi/4}\,|1_L\rangle\,.$$

To circumvent the Eastin–Knill theorem, several approaches have been developed. For example, one may use continuous-variable systems, switch between different codes via suitable quantum operations, or employ *magic state distillation*.

In the following, we will describe the procedure of magic state distillation and explain how one can achieve universal quantum computation using a combination of the Steane code and the Reed–Muller code.

5.6. **Magic State Distillation.** We now discuss the procedure of *magic state distillation*. As we have seen, there is no transversal implementation of the $\mathbf{T}$ gate for the Steane code. Therefore, if we attempt to construct a logical $\overline{\mathbf{T}}$ and apply it to a logical state $\overline{|\psi\rangle}$, the resulting state $\overline{\mathbf{T}}(\overline{|\psi\rangle})$ may contain errors that the Steane code cannot correct.

From now on, for simplicity, we will denote the (possibly faulty) prepared state $\overline{\mathbf{T}}(\overline{|\psi\rangle})$ simply by $\mathbf{T}\,|\psi\rangle$.

The idea behind magic state distillation is the following: rather than trying to apply a fault-tolerant logical $\mathbf{T}$ gate directly, we instead prepare several *faulty* copies of $\mathbf{T}\,|\psi\rangle$, and

then use a special distillation protocol to combine them into a new state which is "better". In this process, we make use of the 15-qubit Reed–Muller code and the fact that it *does* support a transversal implementation of the $\mathbf{T}$ gate.

To make precise the notion of a "better", we use the *fidelity* between quantum states. For a mixed state ρ and a pure state $|\phi\rangle$, the fidelity is defined as

$$F(\rho, |\phi\rangle) := \langle\phi|\,\rho\,|\phi\rangle\,.$$

In our context, ρ will be a noisy (non-transversal) preparation of the state $\mathbf{T}\,|\psi\rangle$, and $|\phi\rangle = \mathbf{T}\,|\psi\rangle$ will be the ideal state. Note that if $\rho = |\phi\rangle\langle\phi|$ is pure and equal to the target state, then $F(\rho, |\phi\rangle) = 1$.

We assume that the initial noisy state ρ is "close" to $\mathbf{T}\,|\psi\rangle$, meaning that $F(\rho, \mathbf{T}\,|\psi\rangle)$ is close to 1. The goal of magic state distillation is to produce a new state $\tilde{\rho}$ such that

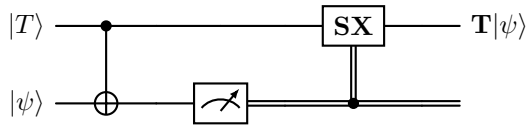
$$F(\tilde{\rho}, \mathbf{T}\,|\psi\rangle) > F(\rho, \mathbf{T}\,|\psi\rangle),$$

i.e., $\tilde{\rho}$ is *closer* to the ideal state than the original ρ.

Before describing the magic state distillation protocol, we first explain what *magic states* are. Let us denote the one-qubit state $\mathbf{T}\,|+\rangle$ by $|T\rangle$. Hence

$$|T\rangle = \frac{|0\rangle + \omega\,|1\rangle}{\sqrt{2}}, \qquad \text{where } \omega = e^{i\pi/4}.$$

There exists a quantum circuit, called the *gate teleportation circuit*, which uses the state $|T\rangle$ as a resource to implement the operation $\mathbf{T}\,|\psi\rangle$ for any one-qubit input $|\psi\rangle$. The circuit is shown below:



Exercise 5.22. ★ *Verify that the above circuit indeed produces the state $\mathbf{T}\,|\psi\rangle$ on the output qubit.*

Exercise 5.23. ★ *Check that if we start with $\mathbf{Z}\,|T\rangle$ in the above circuit, we get the output $\mathbf{ZT}\,|\psi\rangle$.*

Exercise 5.24. ★ *Let Z_T be the operator defined as $Z_T := \omega^*\mathbf{SX}$, and define the state $|T^c\rangle := \mathbf{Z}\,|T\rangle$.*

Show the following:

(1) The states $|T\rangle$ and $|T^c\rangle$ form a basis of the one-qubit Hilbert space $\mathcal{H} = \mathbb{C}^2$.

(2) The set of operators

$$\big\{|T\rangle\langle T|\,,\ |T\rangle\langle T^c|\,,\ |T^c\rangle\langle T|\,,\ |T^c\rangle\langle T^c|\big\}$$

forms a basis of $\mathcal{B}(\mathcal{H})$, the space of all linear operators on $\mathbb{C}^2$.

(3) The operator Z_T acts as

$$Z_T\,|T\rangle = |T\rangle\,, \qquad Z_T\,|T^c\rangle = -\,|T^c\rangle\,.$$

Note that Z_T is a Clifford operator. Because of the property established in the previous exercise—that $|T\rangle$ is an eigenstate of the Clifford operator Z_T—the state $|T\rangle$ is referred to as a *magic state*. Magic states play a fundamental role in enabling non-Clifford operations.

We will now describe the magic state distillation protocol and see how the magic state $|T\rangle$ is used as a key resource to achieve universal quantum computation.

In view of the gate teleportation circuit, it is enough to consider the case where $|\psi\rangle = |+\rangle$. Let ρ be a noisy (non-transversal) preparation of the magic state $|T\rangle$, and as before, assume that the fidelity $F(\rho, |T\rangle)$ is close to 1.

We now apply a "twirling" channel to ρ defined by

$$\mathcal{E}(\rho) := \frac{1}{2}\rho + \frac{1}{2} Z_T \rho Z_T^\dagger.$$

Lemma 5.25. *Let Z_T be the Clifford operator with $Z_T |T\rangle = |T\rangle$, and let*

$$\mathcal{E}(\rho) := \tfrac{1}{2}\rho + \tfrac{1}{2}\, Z_T \rho Z_T^\dagger.$$

Then the fidelity with the magic state is preserved under $\mathcal{E}$:

$$F(\rho, |T\rangle) \;=\; F(\mathcal{E}(\rho), |T\rangle) \;=\; \langle T|\, \rho\, |T\rangle\,.$$

Proof. We have $F(\rho, |T\rangle) = \langle T|\, \rho\, |T\rangle$. Using $Z_T |T\rangle = |T\rangle$ and unitarity of Z_T we get $Z_T^\dagger |T\rangle = |T\rangle$. Hence

$$\langle T|\; Z_T \rho Z_T^\dagger \;|T\rangle = \langle T|\, \rho\, |T\rangle\,.$$

Therefore,

$$F(\mathcal{E}(\rho), |T\rangle) = \langle T|\, \mathcal{E}(\rho)\, |T\rangle = \tfrac{1}{2}\, \langle T|\, \rho\, |T\rangle + \tfrac{1}{2}\, \langle T|\, Z_T \rho Z_T^\dagger\, |T\rangle = \langle T|\, \rho\, |T\rangle = F(\rho, |T\rangle).$$

□

Using Exercise 5.24, the state ρ can be written in the magic basis as

$$\rho = a\, |T\rangle\langle T| \;+\; b\, |T\rangle\langle T^c| \;+\; c\, |T^c\rangle\langle T| \;+\; d\, |T^c\rangle\langle T^c|\,,$$

for some $a, b, c, d \in \mathbb{C}$. Applying the twirling channel from above and using Exercise 5.24, one checks that

$$\mathcal{E}(\rho) = a\, |T\rangle\langle T| + d\, |T^c\rangle\langle T^c|\,.$$

Since $\mathcal{E}(\rho)$ is a valid quantum state, we may write

$$\mathcal{E}(\rho) = (1-p)\, |T\rangle\langle T| + p\, |T^c\rangle\langle T^c|$$

for some $p \in [0, 1]$. Using Lemma 5.25 and the assumption that $F(\rho, |T\rangle)$ is close to 1, one sees that p must be small.

The magic state distillation protocol is now given by the following steps:

(1) Start with 15 copies of $\mathcal{E}(\rho) = (1-p)\, |T\rangle\langle T| + p\, |T^c\rangle\langle T^c|$, where p is small.
(2) Prepare the encoded state $|+\rangle_L$ of the 15-qubit Reed–Muller code.
(3) Apply the gate teleportation circuit to each physical qubit of the 15-qubit Reed–Muller block, using $\mathcal{E}(\rho)$ in place of $|T\rangle$.
(4) Perform error correction for the 15-qubit Reed–Muller code.
(5) Decode the output to obtain a single-qubit state $\tilde{\rho}$.

This completes the distillation procedure.

Exercise 5.26. ★ *Show that the fidelity improves after distillation, i.e.,*

$$F(\tilde{\rho}, |T\rangle) \;>\; F(\rho, |T\rangle),$$

provided that p is sufficiently small.

Notes

- Readers are encouraged to consult Nielsen and Chuang 2010, Chapter 4 for a more detailed discussion of quantum gates.

- The presentation of transversal gates and magic state distillation is influenced by *math.uwaterloo.ca* 2024. For a more detailed account of magic state distillation, the reader is referred to Fujii 2015, Section 2.8.

References for Lecture 11 & 12.

Boykin, P.Oscar et al. (2000). "A new universal and fault-tolerant quantum basis". In: *Information Processing Letters* 75.3, pp. 101–107. ISSN: 0020-0190. DOI: https://doi.org/10.1016/S0020-0190(00)00084-3. URL: https://www.sciencedirect.com/science/article/pii/S0020019000000843.

Fujii, Keisuke (Dec. 2015). *Quantum computation with topological codes*. en. 1st ed. SpringerBriefs in Mathematical Physics. Singapore, Singapore: Springer.

math.uwaterloo.ca (2024). https://www.math.uwaterloo.ca/~wcleung/qic890-w2024.html. [Accessed 16-06-2026].

Nielsen, Michael A. and Isaac L. Chuang (2010). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press.

URL of ICTP SAIFR (2022). https://www.ictp-saifr.org/wp-content/uploads/2022/11/ICTP_SAIFR_D1-L2.pdf. [Accessed 19-06-2026].

IAI, TCG CREST, Kolkata, India
Email address: sayan2008@gmail.com