# Quantum Information
# Chapter 7. Quantum Error Correction

John Preskill
Institute for Quantum Information and Matter
California Institute of Technology

Updated March 2026

For further updates and additional chapters, see: https://preskill.caltech.edu/ph219/

Please send corrections to preskill@caltech.edu

# Contents

# Preface

This is the 7th chapter of my book *Quantum Information*, based on the course I have been teaching at Caltech since 1997. An early version of this chapter has been available on the course website since 1999, but this version is substantially revised and expanded.

This is a working draft of Chapter 7, which I will continue to update. See the URL on the title page for further updates and drafts of other chapters. Please send an email to preskill@caltech.edu if you notice errors.

Eventually, the complete book will be published by Cambridge University Press. I hesitate to predict the publication date — they have been far too patient with me.

# 7

# Quantum Error Correction

## 7.1 A Quantum error-correcting code

In our study of quantum algorithms, we have found persuasive evidence that a quantum computer would have extraordinary power. But will quantum computers really work? Will we ever be able to build and operate them?

To do so, we must rise to the challenge of protecting quantum information from errors. As we have already noted in Chapter 1, there are several aspects to this challenge. A quantum computer will inevitably interact with its surroundings, resulting in decoherence and hence in the decay of the quantum information stored in the device. Unless we can successfully combat decoherence, our computer is sure to fail. And even if we were able to prevent decoherence by perfectly isolating the computer from the environment, errors would still pose grave difficulties. Quantum gates (in contrast to classical gates) are unitary transformations chosen from a continuum of possible values. Thus quantum gates cannot be implemented with perfect accuracy; the effects of small imperfections in the gates will accumulate, eventually leading to a serious failure in the computation. Any effective strategem to prevent errors in a quantum computer must protect against small unitary errors in a quantum circuit, as well as against decoherence.

In this and the next chapter we will see how clever encoding of quantum information can protect against errors (in principle). This chapter will present the theory of quantum error-correcting codes. We will learn that quantum information, suitably encoded, can be deposited in a quantum memory, exposed to the ravages of a noisy environment, and recovered without damage (if the noise is not too severe). Then in Chapter 8, we will extend the theory in two important ways. We will see that the recovery procedure can work effectively even if occasional errors occur during recovery. And we will learn how to *process* encoded information, so that a quantum *computation* can be executed successfully despite the debilitating effects of decoherence and faulty quantum gates.

A quantum error-correcting code (QECC) can be viewed as a mapping of $k$ qubits (a Hilbert space of dimension $2^k$) into $n$ qubits (a Hilbert space of dimension $2^n$), where $n > k$. The $k$ qubits are the "logical qubits" or "encoded qubits" that we wish to protect from error. The additional $n-k$ qubits allow us to store the $k$ logical qubits in a redundant fashion, so that the encoded information is not easily damaged.

We can better understand the concept of a QECC by revisiting an example that was introduced in Chapter 1, Shor's code with $n = 9$ and $k = 1$. We can characterize the code by specifying two basis states for the code subspace; we will refer to these basis

states as $|\bar{0}\rangle$, the "logical zero" and $|\bar{1}\rangle$, the "logical one." They are

$$|\bar{0}\rangle = [\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)]^{\otimes 3},$$
$$|\bar{1}\rangle = [\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)]^{\otimes 3}; \tag{7.1}$$

each basis state is a 3-qubit cat state, repeated three times. As you will recall from the discussion of cat states in Chapter 4, the two basis states can be distinguished by the 3-qubit observable $\boldsymbol{\sigma}_x^{(1)} \otimes \boldsymbol{\sigma}_x^{(2)} \otimes \boldsymbol{\sigma}_x^{(3)}$ (where $\boldsymbol{\sigma}_x^{(i)}$ denotes the Pauli matrix $\boldsymbol{\sigma}_x$ acting on the ith qubit); we will use the notation $\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3$ for this operator. (There is an implicit $\boldsymbol{I} \otimes \boldsymbol{I} \otimes \cdots \otimes \boldsymbol{I}$ acting on the remaining qubits that is suppressed in this notation.) The states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are eigenstates of $\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3$ with eigenvalues $+1$ and $-1$ respectively. But there is no way to distinguish $|\bar{0}\rangle$ from $|\bar{1}\rangle$ (to gather any information about the value of the logical qubit) by observing any one or two of the qubits in the block of nine. In this sense, the logical qubit is encoded *nonlocally*; it is written in the nature of the entanglement among the qubits in the block. This nonlocal property of the encoded information provides protection against noise, if we assume that the noise is local (that it acts independently, or nearly so, on the different qubits in the block).

Suppose that an unknown quantum state has been prepared and encoded as $a|\bar{0}\rangle + b|\bar{1}\rangle$. Now an error occurs; we are to diagnose the error and reverse it. How do we proceed? Let us suppose, to begin with, that a single bit flip occurs acting on one of the first three qubits. Then, as discussed in Chapter 1, the location of the bit flip can be determined by measuring the two-qubit operators

$$\boldsymbol{Z}_1\boldsymbol{Z}_2. \quad \boldsymbol{Z}_2\boldsymbol{Z}_3. \tag{7.2}$$

The logical basis states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are eigenstates of these operators with eigenvalue 1. But flipping any of the three qubits changes these eigenvalues. For example, if $\boldsymbol{Z}_1\boldsymbol{Z}_2 = -1$ and $\boldsymbol{Z}_2\boldsymbol{Z}_3 = 1$, then we infer that the first qubit has flipped relative to the other two. We may recover from the error by flipping that qubit back.

It is crucial that our measurement to diagnose the bit flip is a collective measurement on two qubits at once — we learn the value of $\boldsymbol{Z}_1\boldsymbol{Z}_2$, but we must not find out about the separate values of $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$, for to do so would damage the encoded state. How can such a collective measurement be performed? In fact we can carry out collective measurements if we have a quantum computer that can execute controlled-NOT gates. We first introduce an additional "ancilla" qubit prepared in the state $|0\rangle$, then execute the quantum circuit

– Figure –

and finally measure the ancilla qubit. If the qubits 1 and 2 are in a state with $\boldsymbol{Z}_1\boldsymbol{Z}_2 = -1$ (either $|0\rangle_1|1\rangle_2$ or $|1\rangle_1|0\rangle_2$), then the ancilla qubit will flip once and the measurement outcome will be $|1\rangle$. But if qubits 1 and 2 are in a state with $\boldsymbol{Z}_1\boldsymbol{Z}_2 = 1$ (either $|0\rangle_1|0\rangle_2$ or $|1\rangle_1|1\rangle_2$), then the ancilla qubit will flip either twice or not at all, and the measurement

outcome will be $|0\rangle$. Similarly, the two-qubit operators

$$\begin{aligned} \boldsymbol{Z}_4\boldsymbol{Z}_5, \qquad & \boldsymbol{Z}_7\boldsymbol{Z}_8, \\ \boldsymbol{Z}_5\boldsymbol{Z}_6, \qquad & \boldsymbol{Z}_8\boldsymbol{Z}_9, \end{aligned} \tag{7.3}$$

can be measured to diagnose bit flip errors in the other two clusters of three qubits.

A three-qubit code would suffice to protect against a single bit flip. The reason the 3-qubit clusters are repeated three times is to protect against phase errors as well. Suppose now that a phase error

$$|\psi\rangle \to \boldsymbol{Z}_i|\psi\rangle \tag{7.4}$$

occurs acting on one of the nine qubits. We can diagnose in which cluster the phase error occurred by measuring the two six-qubit observables

$$\begin{aligned} &\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3\boldsymbol{X}_4\boldsymbol{X}_5\boldsymbol{X}_6, \\ &\boldsymbol{X}_4\boldsymbol{X}_5\boldsymbol{X}_6\boldsymbol{X}_7\boldsymbol{X}_8\boldsymbol{X}_9. \end{aligned} \tag{7.5}$$

The logical basis states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are both eigenstates with eigenvalue one of these observables. A phase error acting on any one of the qubits in a particular cluster will change the value of $\boldsymbol{XXX}$ in that cluster relative to the other two; the location of the change can be identified by measuring the observables in eq.(7.5). Once the affected cluster is identified, we can reverse the error by applying $\boldsymbol{Z}$ to any one of the qubits in that cluster.

How do we measure the six-qubit observable $\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3\boldsymbol{X}_4\boldsymbol{X}_5\boldsymbol{X}_6$? Notice that if its control qubit is initially in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and its target is an eigenstate of $\boldsymbol{X}$ (that is, NOT) then a controlled-NOT acts according to

$$\text{CNOT} : \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |x\rangle \to \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x|1\rangle) \otimes |x\rangle; \tag{7.6}$$

it acts trivially if the target is the $\boldsymbol{X} = 1$ ($x = 0$) state, and it flips the control if the target is the $\boldsymbol{X} = -1$ ($x = 1$) state. To measure a product of $\boldsymbol{X}$'s, then, we execute the circuit

– Figure –

and then measure the ancilla in the $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ basis.

We see that a single error acting on any one of the nine qubits in the block will cause no irrevocable damage. But if two bit flips occur in a single cluster of three qubits, then the encoded information *will* be damaged. For example, if the first two qubits in a cluster both flip, we will misdiagnose the error and attempt to recover by flipping the third. In all, the errors, together with our mistaken recovery attempt, apply the operator $\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3$ to the code block. Since $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are eigenstates of $\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3$ with distinct eigenvalues, the effect of two bit flips in a single cluster is a *phase error* in the encoded qubit:

$$\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3 : a|\bar{0}\rangle + b|\bar{1}\rangle \to a|\bar{0}\rangle - b|\bar{1}\rangle. \tag{7.7}$$

The encoded information will also be damaged if phase errors occur in two different

clusters. Then we will introduce a phase error into the third cluster in our misguided attempt at recovery, so that altogether $\boldsymbol{Z}_1\boldsymbol{Z}_4\boldsymbol{Z}_7$ will have been applied, which flips the encoded qubit:

$$\boldsymbol{Z}_1\boldsymbol{Z}_4\boldsymbol{Z}_7 : a|\bar{0}\rangle + b|\bar{1}\rangle \to a|\bar{1}\rangle + b|\bar{0}\rangle. \tag{7.8}$$

If the likelihood of an error is small enough, and if the errors acting on distinct qubits are not strongly correlated, then using the nine-qubit code will allow us to preserve our unknown qubit more reliably than if we had not bothered to encode it at all. Suppose, for example, that the environment acts on each of the nine qubits, independently subjecting it to the depolarizing channel described in Chapter 3, with error probability $p$. Then a bit flip occurs with probability $\frac{2}{3}p$, and a phase flip with probability $\frac{2}{3}p$. (The probability that both occur is $\frac{1}{3}p$). We can see that the probability of a phase error affecting the logical qubit is bounded above by $3 \cdot \binom{3}{2} \cdot \left(\frac{2}{3}p\right)^2 = 4p^2$, and the probability of a bit flip error is bounded above by $3 \cdot 3 \cdot \binom{3}{2} \cdot \left(\frac{2}{3}p\right)^2 = 4p^2 = 12p^2$. The total error probability is no worse than $16p^2$; this is an improvement over the error probability $p$ for an unprotected qubit, provided that $p < 1/16$. (As we'll see in §??, the asymmetry between $\boldsymbol{X}$ and $\boldsymbol{Z}$ errors is actually illusory; the code provides equally good protection against both.)

Of course, in this analysis we have implicitly assumed that encoding, decoding, error syndrome measurement, and recovery are all performed flawlessly. In Chapter 8 we will examine the more realistic case in which errors occur during these operations.

## 7.2 Conditions for quantum error correction

In our discussion of error recovery using the nine-qubit code, we have assumed that each qubit undergoes either a bit-flip error or a phase-flip error (or both). This is not a realistic model for the errors, and we must understand how to implement quantum error correction under more general conditions.

Consider a quantum memory, the system $S$ with Hilbert space $\mathcal{H}_S$, which is subjected to noise, described by a CPTP map $\mathcal{N}$. As we discussed in depth in Chapter 3, $\mathcal{N}$ has a dilation, an isometric (inner-product preserving) map $\boldsymbol{U}$ taking $S$ to $SE$, where $E$ is the system's environment. If $\{\boldsymbol{E}_a\}$ is a set of operators which span the space of linear operators taking $S$ to $S$, then $\boldsymbol{U}$ can be expanded as

$$\boldsymbol{U}_{S\to SE} : |\psi\rangle_S \mapsto \sum_a \boldsymbol{E}_a|\psi\rangle_S \otimes |\varphi_a\rangle_E. \tag{7.9}$$

Here the vectors $\{|\varphi\rangle_a\}$ are *not* assumed to be normalized or mutually orthogonal; hence eq.(7.9) entails no loss of generality. In the case where all the $|\varphi_a\rangle$ are parallel, it describes a *unitary error*, in which a pure state of $S$ remains pure but is subjected to an unknown rotation in Hilbert space. In the case where $\{|\varphi_a\rangle\}$ includes mutually orthogonal vectors, it describes strong decoherence, in which an initially pure state of $S$ becomes mixed due to its entanglement with the environment.

Now we would like to reverse the damage caused by the error by applying a recovery operation $\mathcal{R}$, which is also a CPTP map, and therefore has an isometric dilation $\boldsymbol{V}$ taking $S$ to $SA$, where $A$ is an *ancilla*, an extra system under our control. Unfortunately, successful recovery will not be possible for arbitrary errors of the form eq.(7.9), if we do not know what error occurred, and the environment $E$ is beyond our control. At best we will only be able to reverse errors of a restricted type.

Let's consider a restricted class of noise operations, where the error lies within the

space spanned by a subset $\mathcal{E} \subset \{\boldsymbol{E}_a\}$. We say that the errors in $\mathcal{E}$ are *correctable* if we can recover from any error of the restricted form

$$\tilde{\boldsymbol{U}}_{S\to SE} : |\psi\rangle_S \mapsto \sum_{\boldsymbol{E}_a\in\mathcal{E}} \boldsymbol{E}_a|\psi\rangle_S \otimes |\varphi_a\rangle_E. \tag{7.10}$$

Our choice of $\mathcal{E}$ will be motivated by the physical properties of the noise. For example, if $S$ is an $n$-qubit system, and the noise is weak and acts nearly independently on the qubits, then we may anticipate that any $\boldsymbol{E}_a$ acting nontrivially on a large portion of the $n$ qubits occurs in the sum eq.(7.9) with only a very small amplitude $\||\varphi_a\rangle\| \ll 1$. In that event we might choose $\mathcal{E}$ to contain only operators of low weight (those acting nontrivially on a relatively small number of qubits). If we can correct all the errors in $\mathcal{E}$, then we will be able to nearly reverse the effect of $\boldsymbol{U}$, with the small residual error arising only from $\boldsymbol{E}_a \notin \mathcal{E}$ in eq.(7.9).

But even if we restrict the errors to the correctable set $\mathcal{E}$, we will not be able to protect arbitrary pure states in $\mathcal{H}_S$ from error. At best we will be able to protect only the states lying within a *code subspace* $\mathcal{H}_{\text{code}} \subset \mathcal{H}_S$. If $\mathcal{H}_{\text{code}}$ and $\mathcal{E}$ are appropriately chosen, then we can construct a recovery isometry $\boldsymbol{V}$ (a dilation of the CPTP map $\mathcal{R}$) such that

$$\boldsymbol{V}_{S\to SA} : \boldsymbol{E}_a|\psi\rangle_S \mapsto |\psi\rangle_S \otimes |\chi_a\rangle_A \tag{7.11}$$

for all $|\psi\rangle_S \in \mathcal{H}_{\text{code}}$ and $\boldsymbol{E}_a \in \mathcal{E}$, and for some set of vectors $\{|\chi_a\rangle_A \in \mathcal{H}_A\}$. It then follows that

$$|\psi\rangle_S \overset{\tilde{\boldsymbol{U}}}{\mapsto} \sum_{\boldsymbol{E}_a\in\mathcal{E}} \boldsymbol{E}_a|\psi\rangle_S \otimes |\varphi_a\rangle_E \overset{\boldsymbol{V}}{\mapsto} |\psi\rangle_S \otimes \left(\sum_a |\varphi_a\rangle_E \otimes |\chi_a\rangle_A\right). \tag{7.12}$$

In general, the noise map $\tilde{\boldsymbol{U}}$ generates entanglement between $S$ and $E$. We cannot undo that $E$ became entangled with something, because we do not control $E$. But what we can do, under suitable conditions, is transform entanglement of $E$ with $S$ into entanglement of $E$ with $A$ while purifying $S$ as in eq.(7.12).

For the recovery operation to exist, the code space $\mathcal{H}_{\text{code}}$ and correctable error set $\mathcal{E}$ must obey a necessary condition which just follows from the requirement that $\boldsymbol{V}$ is isometric. For any normalized pure state $|\psi\rangle_S$ in the code space, eq.(7.11) implies that

$$\langle\psi|\boldsymbol{E}_b^\dagger \boldsymbol{E}_a|\psi\rangle = \langle\psi|\psi\rangle\langle\chi_b|\chi_a\rangle = \langle\chi_b|\chi_a\rangle := C_{ba}; \tag{7.13}$$

here $C_{ba}$ is a nonnegative Hermitian matrix, not depending on $|\psi\rangle$. It is sometimes convenient to write this condition differently. If $\{|\bar{i}\rangle\}$ is an orthonormal basis for $\mathcal{H}_{\text{code}}$, and eq.(7.13) holds for any state $|\psi\rangle = \sum_i \psi_i|\bar{i}\rangle$ in the code space, then

$$\sum_{i,j} \psi_j^*\psi_i\langle\bar{j}|\boldsymbol{E}_b^\dagger \boldsymbol{E}_a|\bar{i}\rangle = C_{ba}\sum_i \psi_i^*\psi_i, \tag{7.14}$$

and hence

$$\langle\bar{j}|\boldsymbol{E}_b^\dagger \boldsymbol{E}_a|\bar{i}\rangle = \delta_{ji}C_{ba}. \tag{7.15}$$

Expressed in terms of

$$\boldsymbol{\Pi} = \sum_i |\bar{i}\rangle\langle\bar{i}|, \tag{7.16}$$

the orthogonal projector onto the code space, eq.(7.15) becomes

$$\mathbf{\Pi}\boldsymbol{E}_b^\dagger \boldsymbol{E}_a \mathbf{\Pi} = C_{ba}\mathbf{\Pi}. \tag{7.17}$$

Eq.(7.17), which must hold for each pair of error operators $\boldsymbol{E}_a$ and $\boldsymbol{E}_b$ in the correctable set $\mathcal{E}$, is called the *quantum error correction condition* or *Knill-Laflamme condition.*

In fact, eq.(7.17) is a sufficient as well as necessary condition for the existence of a recovery operator protecting the code space against the errors in $\mathcal{E}$. Since $C_{ba}$ is Hermitian and nonnegative, we can choose a basis $\tilde{\mathcal{E}} = \{\tilde{\boldsymbol{E}}_a\}$ in which it is diagonal: $\tilde{C}_{ba} = c_a\delta_{ba}$ and $c_a \geq 0$; hence

$$\tilde{\boldsymbol{U}}_{S\to SE} : |\psi\rangle_S \mapsto \sum_{\tilde{E}_a\in\tilde{\mathcal{E}}} \tilde{\boldsymbol{E}}_a|\psi\rangle_S \otimes |\tilde{\varphi}_a\rangle_E, \tag{7.18}$$

where

$$\mathbf{\Pi}\tilde{\boldsymbol{E}}_b^\dagger \tilde{\boldsymbol{E}}_a \mathbf{\Pi} = c_a\delta_{ba}\mathbf{\Pi}. \tag{7.19}$$

Since we also have the freedom to rescale the error operators (while performing a compensating rescaling of the vectors $\{|\tilde{\varphi}_a\rangle_E\}$), we may assume that each $c_a$ is either 0 or 1.

Let's first consider the case of a *nondegenerate code* in which $c_a > 0$ for each $a$, and hence (after a suitable rescaling, if necessary)

$$\mathbf{\Pi}\tilde{\boldsymbol{E}}_b^\dagger \tilde{\boldsymbol{E}}_a \mathbf{\Pi} = \delta_{ba}\mathbf{\Pi}. \tag{7.20}$$

For a nondegenerate code, all of the "error subspaces" $\tilde{\boldsymbol{E}}_a\mathcal{H}_{\text{code}}$ are mutually orthogonal and hence perfectly distinguishable. If a correctable error $\tilde{\boldsymbol{E}}_a \in \mathcal{E}$ acts on the code space, $\tilde{\boldsymbol{E}}_a$ can be unambiguously identified by performing a suitable orthogonal measurement. Furthermore, the error $\tilde{\boldsymbol{E}}_a$ isometrically maps the code space to the corresponding error space. Therefore, once $\tilde{\boldsymbol{E}}_a$ is identified, we can apply a unitary map taking the error space back to the code space, correcting the error.

To be more explicit, the orthogonal projectors

$$\{\mathbf{\Pi}_a = \tilde{\boldsymbol{E}}_a\mathbf{\Pi}\tilde{\boldsymbol{E}}_a^\dagger\} \tag{7.21}$$

satisfying

$$\mathbf{\Pi}_a\mathbf{\Pi}_b = \delta_{ab}\mathbf{\Pi}_a \tag{7.22}$$

project onto the error subspaces. These projectors may not provide a complete partition of $\mathcal{H}_S$, but we can add one additional projector, onto the orthogonal complement of $\oplus_a\tilde{\boldsymbol{E}}_a\mathcal{H}_{\text{code}}$, to define an orthogonal measurement. The measurement outcome $\mathbf{\Pi}_a$ provides an *error syndrome* pointing to the error space $\tilde{\boldsymbol{E}}_a\mathcal{H}_{\text{code}}$, where this outcome occurs with probability $\text{prob}(a) = \langle\tilde{\chi}_a|\tilde{\chi}_a\rangle$ if the noise map is given by eq.(7.18) and the error operators obey eq.(7.20); once the outcome $a$ is known, a unitary transformation $\boldsymbol{U}_a$ applied to $\mathcal{H}_S$ corrects the error. The transformation $\boldsymbol{U}_a$ is chosen so that it acts as $\mathbf{\Pi}\tilde{\boldsymbol{E}}_a^\dagger$ on the subspace $\tilde{\boldsymbol{E}}_a\mathcal{H}_{\text{code}}$, while its action on the complement of $\tilde{\boldsymbol{E}}_a\mathcal{H}_{\text{code}}$ may be chosen arbitrarily.

This argument shows that recovery from error is possible if eq.(7.17) is satisfied and the code is nondegenerate. But what if the code is degenerate — that is, what if $\tilde{\boldsymbol{E}}_a\mathbf{\Pi} = 0$ for some of the error operators? This happens if our basis of correctable errors is overcomplete, so that there are linear combinations of elements of $\mathcal{E}$ which annihilate

the code space. In that case, we may simply eliminate these redundant error operators from our set of correctable errors, replacing our set $\mathcal{E}$ of correctable errors by a smaller set $\tilde{\mathcal{E}} \subset \mathcal{E}$.

We have already encountered an example of a degenerate code — the 9-qubit code, in which phase errors acting on the same cluster of three qubits affect the code subspace in the same way (*e.g.*, $\boldsymbol{Z}_1|\psi\rangle = \boldsymbol{Z}_2|\psi\rangle$, where $|\psi\rangle \in \mathcal{H}_{\rm code}$). This means that no syndrome measurement can distinguish whether error $\boldsymbol{Z}_1$ or $\boldsymbol{Z}_2$ occurred, and correspondingly, there is an error $\boldsymbol{Z}_1 - \boldsymbol{Z}_2$ which annihilates the code space. As this "error" occurs with zero amplitude, we can eliminate it from the sum over $\tilde{\boldsymbol{E}}_a$ in eq.(7.18) without changing anything. Once all such zero-amplitude errors have been eliminated, we can measure the error syndrome and correct the error just as in the nondegenerate case. Therefore, the error correction condition eq.(7.17) is sufficient as well as necessary for the existence of a recovery map that reverses the error, whether or not the code is degenerate.

We have described this recovery procedure as a two step process, in which the error syndrome is measured in the first step, and an error-correcting unitary operator conditioned on the outcome of the syndrome measurement is performed in the second step. But in order to recover, it is not necessary to perform a measurement that amplifies the error syndrome to the classical world. Instead, we may sum coherently over the possible syndromes, obtaining an isometric recovery map $\boldsymbol{V}$. Acting on the span of $\{\tilde{\boldsymbol{E}}_a \mathcal{H}_{\rm code}, \tilde{\boldsymbol{E}}_a \in \tilde{\mathcal{E}}\}$, this isometric map is

$$\boldsymbol{V} : |\varphi\rangle_S \mapsto \sum_b \boldsymbol{\Pi} \tilde{\boldsymbol{E}}_b^\dagger |\varphi\rangle_S \otimes |b\rangle_A \tag{7.23}$$

where $\{|b\rangle_A\}$ is an orthonormal basis for the ancilla (as follows from eq.(7.13) if the error operators obey eq.(7.20)). This recovery map can correct any error of the form eq.(7.18) acting on the code space:

$$\begin{aligned} \boldsymbol{V}\tilde{\boldsymbol{U}} : |\psi\rangle_S &\mapsto \sum_{a,b} \boldsymbol{\Pi} \tilde{\boldsymbol{E}}_b^\dagger \tilde{\boldsymbol{E}}_a \boldsymbol{\Pi} |\psi\rangle \otimes |\tilde{\varphi}_a\rangle_E \otimes |b\rangle_A \\ &= |\psi\rangle_S \otimes \left( \sum_a |\tilde{\varphi}_a\rangle_E \otimes |a\rangle_A \right). \end{aligned} \tag{7.24}$$

Eq.(7.20) ensures that $\boldsymbol{V}$ is an isometry acting on $\mathrm{span}\{\boldsymbol{E}_a \mathcal{H}_{\rm code}\}$, and it can be extended to an isometry acting on all of $\mathcal{H}_S$. If the ancilla is initialized in a standard pure state $|0\rangle_A$, the isometry $\boldsymbol{V}$ can be realized as a unitary transformation mapping $SA$ to itself. A suitably constructed quantum circuit can execute this recovery transformation; no measurement of the syndrome is required.

– Figure –

*Show $|\psi\rangle$ subjected to noise, and then decoded as ancilla is discarded.*

It should be emphasized that the construction of the recovery map $\boldsymbol{V}$ depends only on the code space $\mathcal{H}_{\rm code}$ and on the correctable error set $\mathcal{E}$. Recovery succeeds for any choice of the vectors $\{|\varphi_a\rangle_E\}$ in eq.(7.10); we don't need to know anything further about the error, only that it lies within the span of the error operators in $\mathcal{E}$. In particular, the

code can be protected against both unitary control errors and decoherence errors, as long as errors outside $\mathcal{E}$ occur with negligible amplitude.

Although an orthogonal measurement of the syndrome is not a necessary part of the recovery procedure, the ancilla is absolutely essential. In effect, the noise "heats up" the quantum memory $S$, and the recovery operation cools it back down. The ancilla $A$ is the cold reservoir where this excess heat is deposited.

If we wish to protect the quantum information stored in a quantum memory for a long time, while the memory is continuously subjected to noise, then we should perform the error recovery map many times in succession. Each recovery step makes use of a fresh ancilla, which may be discarded after use. But if only a limited supply of ancilla qubits are available, then the ancilla should be erased before it is recycled for further error recovery steps; that is, the ancilla must be mapped back to the standard initial state $|0\rangle_A$, in effect cooling it to a very low temperature. This erasure is a dissipative process; work is needed to pump the heat from the ancilla back to the environment. Thus, unless we have an unlimited supply of fresh ancilla qubits, error recovery incurs an unavoidable thermodynamic cost. We cannot perform quantum (or classical) error correction for free.

## 7.3 More about the error correction condition

The condition eq.(7.17) is the foundation of the theory of quantum error correction. In this section we will explore further the interpretation and implications of this error correction condition; considering it from complementary vantage points deepens our appreciation of how and why quantum error correction works.

### *7.3.1 Distinguishability of orthogonal code states*

We can protect encoded quantum information against noise only if the noise preserves the perfect distinguishability of orthogonal vectors in the code space. Thus if $\boldsymbol{E}_a$, $\boldsymbol{E}_b$ are any two errors in the correctable set $\mathcal{E}$, and $|\varphi\rangle$, $|\psi\rangle$ are any two orthogonal pure states in $\mathcal{H}_{\text{code}}$, then $\boldsymbol{E}_a|\varphi\rangle$ and $\boldsymbol{E}_b|\psi\rangle$ must also be orthogonal. Otherwise, the recovery isometry cannot possibly map $\boldsymbol{E}_a|\varphi\rangle$ and $\boldsymbol{E}_b|\psi\rangle$ to the orthogonal states $|\varphi\rangle$ and $|\psi\rangle$.

To be concrete, suppose the code space is two-dimensional and let $\{|\bar{0}\rangle, |\bar{1}\rangle\}$ denote an orthonormal basis for $\mathcal{H}_{\text{code}}$. For distinguishability of the basis vectors to be preserved by the correctable errors, we require that $\boldsymbol{E}_a|\bar{0}\rangle \perp \boldsymbol{E}_b|\bar{1}\rangle$ for $\boldsymbol{E}_a, \boldsymbol{E}_b \in \mathcal{E}$, or

$$\langle\bar{1}|\boldsymbol{E}_b^\dagger \boldsymbol{E}_a|\bar{0}\rangle = 0 = \langle\bar{0}|\boldsymbol{E}_b^\dagger \boldsymbol{E}_a|\bar{1}\rangle. \tag{7.25}$$

Distinguishability must also be preserved for the conjugate basis states $\{\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)\}$; therefore

$$0 = (\langle\bar{0}| + \langle\bar{1}|)\,\boldsymbol{E}_b^\dagger \boldsymbol{E}_a\,(|\bar{0}\rangle - |\bar{1}\rangle) = \langle\bar{0}|\boldsymbol{E}_b^\dagger \boldsymbol{E}_a|\bar{0}\rangle - \langle\bar{1}|\boldsymbol{E}_b^\dagger \boldsymbol{E}_a|\bar{1}\rangle \tag{7.26}$$

Putting together eq.(7.25) and (7.26) we see that the error correction condition

$$\langle\bar{j}|\boldsymbol{E}_b^\dagger \boldsymbol{E}_a|\bar{i}\rangle = \delta_{ji} C_{ba}. \tag{7.27}$$

is satisfied for $\boldsymbol{E}_a, \boldsymbol{E}_b \in \mathcal{E}$; thus survival of perfect distinguishability for both the basis states $|\bar{0}\rangle$, $|\bar{1}\rangle$ and the conjugate basis states suffices to ensure that quantum information in the code space can be perfectly protected against the noise.

The noise might deform the code states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ substantially, but that won't necessarily mean that encoded quantum information is irrevocably lost. As long as orthogonal code vectors are mapped to distinguishable (mixed) states of the system, recovery from error is possible.

### 7.3.2 Leakage of information to the environment

In typical laboratory settings, information stored in a quantum memory is fragile because the memory cannot be perfectly isolated from the outside world, and leakage of information to the environment induces irreversible decoherence. A quantum error-correcting code can prevent this leakage, protecting the encoded state.

If the system $S$ is subjected to an error of the form eq.(7.10), then if the initial state $|\psi\rangle_S$ is in the code space $\mathcal{H}_{\text{code}}$ we find by tracing out $S$ that the state of the environment becomes

$$\begin{aligned}\boldsymbol{\rho}_E &= \mathrm{tr}_S\left(\sum_{\boldsymbol{E}_a,\boldsymbol{E}_b\in\mathcal{E}} \boldsymbol{E}_a|\psi\rangle\langle\psi|\boldsymbol{E}_b^\dagger \otimes |\varphi_a\rangle\langle\varphi_b|\right)\\ &= \sum_{\boldsymbol{E}_a,\boldsymbol{E}_b\in\mathcal{E}} \langle\psi|\boldsymbol{E}_b^\dagger\boldsymbol{E}_a|\psi\rangle|\varphi_a\rangle\langle\varphi_b| = \sum_{a,b} C_{ba}|\varphi_a\rangle\langle\varphi_b|,\end{aligned} \tag{7.28}$$

where in the last equality we have used the error correction condition eq.(7.13). When the error correction condition is satisfied, then, the state of the environment is independent of the encoded quantum state. The converse is also true — if $\boldsymbol{\rho}_E$ does not depend on the code state $|\psi\rangle$, then the error correction conditions are satisfied, and the noise is correctable.

### 7.3.3 Decoupling a reference system

We can recast the error correction condition in another useful form by applying the relative state method described in Chapter 3. Let $\{|\bar{i}\rangle_S\}$ denote an orthonormal basis for the code space $\mathcal{H}_{\text{code}} \subset \mathcal{H}_S$. We introduce a reference system $R$, where the dimension $d_R$ of $\mathcal{H}_R$ is the same as the dimension of $\mathcal{H}_{\text{code}}$, and consider the state

$$|\Phi\rangle_{RS} = \frac{1}{\sqrt{d_R}}\sum_i |i\rangle_R \otimes |\bar{i}\rangle_S, \tag{7.29}$$

in which $R$ is maximally entangled with the code space.

Under the noise isometry $\tilde{\boldsymbol{U}}$ in eq.(7.10), this state is mapped to

$$|\Phi'\rangle_{RSE} = \frac{1}{\sqrt{d_R}}\sum_i \sum_{\boldsymbol{E}_a\in\mathcal{E}} |i\rangle_R \otimes \boldsymbol{E}_a|\bar{i}\rangle_S \otimes |\varphi_a\rangle_E. \tag{7.30}$$

A successful recovery isometry $\boldsymbol{V}_{S\to SA}$ maps this state according to

$$|\Phi'\rangle_{RSE} \overset{\boldsymbol{V}}{\mapsto} |\Phi\rangle_{RS} \otimes |\chi\rangle_{EA} \tag{7.31}$$

restoring the maximal entanglement of $R$ and the code space.

The error correction condition eq.(7.15) provides a characterization of the marginal

state of $RE$ after the noise acts. Tracing out the system $S$ yields

$$\boldsymbol{\rho}_{RE} = \mathrm{tr}_S |\Phi'\rangle\langle\Phi'| = \frac{1}{d_R} \sum_{i,j} \sum_{\boldsymbol{E}_a, \boldsymbol{E}_b \in \mathcal{E}} \langle \bar{j} | \boldsymbol{E}_b^\dagger \boldsymbol{E}_a | \bar{i} \rangle \otimes |i\rangle\langle j| \otimes |\varphi_a\rangle\langle\varphi_b|$$
$$= \frac{1}{d_R} \sum_{i,j,a,b} C_{ba} \delta_{ji} \otimes |i\rangle\langle j| \otimes |\varphi_a\rangle\langle\varphi_b| = \frac{1}{d_R} \boldsymbol{I}_R \otimes \boldsymbol{\rho}_E. \tag{7.32}$$

We conclude that for error correction to be possible, the reference system must remain uncorrelated with the environment, in which case we say that the reference system *decouples* from the environment. This statement $\boldsymbol{\rho}_{RE} = \boldsymbol{\rho}_R \otimes \boldsymbol{\rho}_E$ is another way to formulate the requirement that no leakage of information to the environment is allowed.

The decoupling condition eq.(7.32) is equivalent to the Knill-Laflamme error correction eq.(7.17), and provides an alternative perspective on why this condition suffices to ensure that the noise is correctable. Note that the state $|\Phi'\rangle_{RSE}$ provides a purification of the maximally mixed state $\boldsymbol{\rho}_R$. If $R$ is uncorrelated with $E$, then $R$ must be maximally entangled with some subsystem of $S$. Therefore a recovery operator acting on $S$ exists which maps this subsystem back to the code space, restoring $|\Phi\rangle_{RS}$. (For a more explicit construction, see the discussion of erasure correction in §7.3.4, and see also Chapter 10.)

To prevent misunderstanding, we emphasize that in general the recovery map depends on the noise channel. So, for example, if recovery map $\mathcal{R}_1$ corrects noise channel $\mathcal{N}_1$ and recovery map $\mathcal{R}_2$ corrects noise channel $\mathcal{N}_2$, that's not enough to guarantee correctability if the channel could be either one of $\mathcal{N}_1$ or $\mathcal{N}_2$, and we don't know which. For example, if the noise channel is a unitary transformation acting on $S$, then no channel environment is needed and decoupling is trivially satisfied, even if we take the "code space" to be all of $S$. Indeed, if the noise is the unitary $U_1$ we can recovery with $U_1^\dagger$, and if the noise is the unitary $U_2$ we can recover with $U_2^\dagger$. But that does not mean we can correct a channel in which both of $U_1$ and $U_2$ are possible errors. If, say, those two unitary errors occur with respective probabilities $p_1$ and $p_2$, then the dilation of the noise channel would associate distinct states of the environment with those two possible unitaries, and the decoupling condition could easily fail.

The decoupling condition is quite useful, because it provides a powerful approach to *approximate* quantum error correction. If $\boldsymbol{\rho}_{RE}$ is close to a product state, then a recovery map exists that nearly reverses the error. We'll make heavy use of that idea in Chapter 10.

### *7.3.4 Correcting erasure errors*

When we say that a quantum system has been *erased*, we mean that the system has escaped beyond our reach and can no longer be accessed. Furthermore, we *know* that the system has been lost. Mathematically, we can describe erasure by the CPTP map

$$\boldsymbol{\rho}_S \mapsto |e\rangle\langle e|; \tag{7.33}$$

here $|e\rangle$, called the *erasure symbol*, is a vector orthogonal to the system's Hilbert space $\mathcal{H}_S$. The idea is that we can perform a two-outcome orthogonal measurement which detects whether erasure occurs — the outcome $|e\rangle\langle e|$ confirms erasure while the outcome $I - |e\rangle\langle e| = \Pi_S$ (the projector onto $\mathcal{H}_S$), confirms that $S$ is still available. An isometric

dilation of this erasure map acts according to

$$|\psi\rangle_S \mapsto |e\rangle \otimes |\psi\rangle_E; \tag{7.34}$$

the system is replaced by the erasure symbol while its initial state is transferred to the environment.

Recovery from erasure is of particular interest. Of course, we can't possibly recover if $S$ is completely erased. But suppose instead that $S = BC$ has a decomposition into subsystems, $\mathcal{H}_S = \mathcal{H}_B \otimes \mathcal{H}_C$, where $B$ is erased but $C$ is unharmed. We might anticipate that if the erased subsystem $B$ is not too large, and the code is suitably chosen, then we can recover the (redundantly encoded) protected quantum information using a recovery map that acts on $C$ alone. In effect, if we disregard the superfluous erasure symbol in eq.(7.34), erasure of $B$ maps the state of $S = BC$ to the corresponding state of $C$ and the environment $E$, according to

$$|\psi\rangle_{BC} \mapsto |\psi\rangle_{EC}, \tag{7.35}$$

after which $E$ is traced out. Equivalently, erasure of $B$ is a noise channel mapping $S$ to $C$ defined by tracing out $B$, so that we may just as well regard $B$ as the environment for the dilation of this noise channel.

If a reference system $R$ is maximally entangled with $\mathcal{H}_{\text{code}}$ as in eq.(7.29), then the decoupling condition eq.(7.32) for quantum error correction becomes, for the case of erasure,

$$\boldsymbol{\rho}_{RB} = \frac{1}{d_R} \boldsymbol{I}_R \otimes \boldsymbol{\rho}_B, \tag{7.36}$$

where $B$ is the erased subsystem. We know on general grounds that this decoupling condition suffices to ensure that the erasure of $B$ is correctable. But it will be instructive to construct an explicit recovery map, from which we can extract some additional insights.

The condition eq.(7.36) has interesting implications regarding the structure of the code space. If $\boldsymbol{\rho}_B = \sum_b \lambda_b |b\rangle\langle b|$, where $\{|b\rangle\}$ is an orthonormal basis for $B$, then eq.(7.36) implies that the pure state of $RBC$ has the Schmidt decomposition (across the $RB$-$C$ cut)

$$|\Phi\rangle_{RBC} = \frac{1}{\sqrt{d_R}} \sum_{i,b} \sqrt{\lambda_b}\ |i\rangle_R \otimes |b\rangle_B \otimes |\psi_{bi}\rangle_C, \tag{7.37}$$

where $\{|\psi_{bi}\rangle\}$ is an orthonormal basis for a subspace of $C$ which has dimension $d_B d_R$; here $d_B$ is the the rank of $\boldsymbol{\rho}_B$ (its number of nonvanishing eigenvalues) and $d_R$ is the dimension of $R$. This subspace has a decomposition into subsystems $C_1 C_2$, where $C_1$ has dimension $d_B$ and $C_2$ has dimension $d_R$, such that

$$|\psi_{bi}\rangle_C = \boldsymbol{U}_C \left(|b\rangle_{C_1} \otimes |i\rangle_{C_2}\right) \tag{7.38}$$

for some unitary transformation $\boldsymbol{U}_C$ acting on $C$, and therefore the orthonormal basis states for the code space can be expressed as

$$\begin{aligned} |\bar{i}\rangle_{BC} &= (\boldsymbol{I}_B \otimes \boldsymbol{U}_C) \left( \sum_b \sqrt{\lambda_b}\ |b\rangle_B \otimes |b\rangle_{C_1} \otimes |i\rangle_{C_2} \right) \\ &:= (\boldsymbol{I}_B \otimes \boldsymbol{U}_C)\, |\chi\rangle_{BC_1} \otimes |i\rangle_{C_2}. \end{aligned} \tag{7.39}$$

– Figure –
*Depict the encoding circuit for the code which protects against erasure of B, corresponding to eq.(7.39).*

Eq.(7.39) clarifies how erasure correction can be achieved. The unitary transformation $\boldsymbol{I}_B \otimes \boldsymbol{U}_C^\dagger$, supported only on $C$, can be applied to the code space even if the subsystem $B$ is inaccessible. This transformation maps the code state $|\psi\rangle_S = \sum_i \psi_i |\bar{i}\rangle_S$ to the corresponding state $|\psi\rangle_{C_2} = \sum_i \psi_i |i\rangle_{C_2}$ of $C_2$, tensored with the fixed state $|\chi\rangle_{BC_1}$ of $BC_1$ which does not depend on $|\psi\rangle$. Thus if $B$ is erased, we can apply $\boldsymbol{U}_C^\dagger$, replace the state of $BC_1$ by $|\chi\rangle$, and then apply $\boldsymbol{U}_C$ to recover the undamaged code state.

This observation has a further implication. We say that an operator $\boldsymbol{M}$ acting on $S$ is a *logical operator* if it maps $\mathcal{H}_{\text{code}}$ to $\mathcal{H}_{\text{code}}$. Hence if $\boldsymbol{M}$ is logical it acts on the code's orthonormal basis states as

$$\boldsymbol{M}_S : |\bar{i}\rangle_S \mapsto \sum_j |\bar{j}\rangle_S M_{ji}. \tag{7.40}$$

From eq.(7.39) we see that if erasure of $B$ is correctable, then the logical operator $\boldsymbol{M}_S$ can be realized as an operator supported on $C$ alone:

$$\boldsymbol{M}_S = (\boldsymbol{I}_B \otimes \boldsymbol{U}_C)\,(\boldsymbol{I}_{BC_1} \otimes \boldsymbol{M}_{C_2})\left(\boldsymbol{I}_B \otimes \boldsymbol{U}_C^\dagger\right), \tag{7.41}$$

where

$$\boldsymbol{M}_{C_2} : |i\rangle_{C_2} \mapsto \sum_j |j\rangle_{C_2} M_{ji}. \tag{7.42}$$

We conclude that, if erasure of $B$ can be corrected, then we can apply an arbitrary logical operator to the code space without ever touching the subsystem $B$. This is sometimes called the *cleaning lemma*, because it's possible to "clean" the operator $\boldsymbol{M}$ on the correctable subsystem without altering its action on the code space.

## 7.4 Some general properties of QECC's

We say a quantum code is *binary* if it can be represented in terms of qubits. In a binary code, a code subspace $\mathcal{H}_{\text{code}}$ of dimension $2^k$ is embedded in a system $\mathcal{H}_S$ of dimension $2^n$, where $k$ and $n > k$ are nonnegative integers. We call $n$ the block size or length of the code, and refer to $k$ as the number of encoded or protected qubits. There is actually no need to require the dimensions of $\mathcal{H}_S$ and $\mathcal{H}_{\text{code}}$ to be powers of two (see the exercises); nevertheless we will mostly confine our attention in this chapter to binary coding, which is the simplest case and the case that typically arises in applications to quantum computing.

A convenient basis for the linear operators acting on an $n$-qubit Hilbert space is provided by the $2^{2n}$ Pauli operators

$$\{\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}^{\otimes n}. \tag{7.43}$$

Each Pauli operator, a tensor product of $n$ 2×2 Pauli matrices, can be assigned a *weight*, an integer ranging from 0 to $n$. The weight $w$ is the number of nontrivial Pauli matrices in the tensor product; that is, a weight-$w$ Pauli operator applies the identity $\boldsymbol{I}$ to $n-w$ of the qubits, and applies one of $\{\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}$ to each of the remaining $w$ qubits. Any

$n$-qubit linear operator $\boldsymbol{M}$, even if not a Pauli operator, can also be assigned a weight. Because the Pauli operators are linearly independent, $\boldsymbol{M}$ has a unique expansion as a sum of Pauli operators, and we say $\boldsymbol{M}$ has weight $w$ if a weight-$w$ Pauli operator occurs in that expansion with a nonzero coefficient, while no Pauli operator with weight greater than $w$ occurs in the expansion. Note that if $\boldsymbol{M}_1$ has weight $w_1$ and $\boldsymbol{M}_2$ has weight $w_2$, then the product $\boldsymbol{M}_1\boldsymbol{M}_2$ has weight at most $w_1 + w_2$.

### 7.4.1 Distance

In addition to the length $n$ and the number of encoded qubits $k$, another important parameter characterizing a code is its *distance* $d$. The distance $d$ is the minimum weight of an operator $\boldsymbol{E}$ such that

$$\boldsymbol{\Pi E \Pi} \propto \boldsymbol{\Pi}. \tag{7.44}$$

Since any operator can be expanded as a linear combination of Pauli operators, we may equivalently say that $d$ is the minimum weight for a Pauli operator $\boldsymbol{E}$ satisfying eq.(7.44). We will describe a quantum code with block size $n, k$ encoded qubits, and distance $d$ as an "$[[n, k, d]]$ quantum code." We use the double-bracket notation for quantum codes, to distinguish from the $[n, k, d]$ notation used for classical codes.

We say that a quantum error-correcting code can "correct $t$ errors" if the span of the set $\mathcal{E}$ of correctable errors includes all operators of weight $t$ or less; equivalently, all Pauli operators of weight $\leq t$ are correctable errors. Our definition of distance implies that the criterion for error correction

$$\boldsymbol{\Pi} \boldsymbol{E}_b^\dagger \boldsymbol{E}_a \boldsymbol{\Pi} = C_{ba} \boldsymbol{\Pi}, \tag{7.45}$$

will be satisfied by all operators $\boldsymbol{E}_a$, $\boldsymbol{E}_b$ of weight $t$ or less, provided that $d \geq 2t{+}1$, because the operator $\boldsymbol{E}_b^\dagger \boldsymbol{E}_a$ has weight at most $2t$. Therefore, a quantum error-correcting code with distance $d \geq 2t{+}1$ can correct $t$ errors.

### 7.4.2 Located errors

A distance $d = 2t{+}1$ code can correct $t$ errors, irrespective of the location of the errors in the code block. But in some cases we may know that particular qubits in the code block are the only ones that have been damaged, in which case we speak of *located errors*. A given code can protect more errors if the errors occur at known locations rather than unknown locations.

Located errors might arise because we saw a hammer strike $t < n$ of the $n$ qubits in the code block, while the other $n{-}t$ qubits remained well protected. The erasure errors considered in §7.3.4 may also be regarded as located errors. Perhaps you sent a block of $n$ qubits to me, but $t < n$ of the qubits were lost and never received, while I am confident that the $n{-}t$ qubits that did arrive were well packaged and were received undamaged. This is an erasure error. If I replace the $t$ missing qubits with $t$ qubits in an arbitrary state, the erasure error becomes a located error because I know which of the $t$ qubits in the block of $n$ are the damaged ones.

A QECC with distance $d = t{+}1$ can correct $t$ errors at known locations. In this case, the set $\mathcal{E}$ of errors to be corrected is the set of all Pauli operators with *support* at the $t$ specified locations (each $\boldsymbol{E}_a$ acts trivially on the other $n{-}t$ qubits). But then, for each $\boldsymbol{E}_a$ and $\boldsymbol{E}_b$ in $\mathcal{E}$, the product $\boldsymbol{E}_b^\dagger \boldsymbol{E}_a$ also has weight at most $t$. Therefore, the error

correction criterion is satisfied for all $\boldsymbol{E}_{a,b} \in \mathcal{E}$, provided the code has distance at least $t{+}1$. Thus a QECC that corrects $t$ errors at arbitrary locations can correct $2t$ errors at known locations.

### 7.4.3 Error detection

In some cases we may be satisfied to detect whether an error has occurred, even if we are unable to fully diagnose or reverse the error. A measurement designed for error detection has two possible outcomes: "good" and "bad." If the good outcome occurs, we are assured that the quantum state is undamaged. If the bad outcome occurs, damage has been sustained, and the state should be discarded.

If the error map has its support on the set $\mathcal{E}$ of all Pauli operators of weight up to $t$, and it is possible to make a measurement that correctly diagnoses *whether* an error has occurred, then it is said that we can detect $t$ errors. Error detection is easier than error correction, so a given code can detect more errors than it can correct. In fact, a QECC with distance $d = t{+}1$ can detect $t$ errors.

A code with $d = t{+}1$ has the property

$$\boldsymbol{\Pi} \boldsymbol{E}_a \boldsymbol{\Pi} = C_a \boldsymbol{\Pi} \tag{7.46}$$

for every Pauli operator $\boldsymbol{E}_a$ of weight $t$ or less (where $\boldsymbol{\Pi}$ is the orthogonal projector onto $\mathcal{H}_{\text{code}}$) or

$$\boldsymbol{E}_a |\psi\rangle = C_a |\psi\rangle + |\psi_a^{\perp}\rangle, \tag{7.47}$$

where $|\psi\rangle$ is any vector in $\mathcal{H}_{\text{code}}$ and $|\psi_a^{\perp}\rangle$ is an (unnormalized) vector orthogonal to the code subspace. Therefore, the general error map in which all errors have weight $t$ or less acts on the code state $|\psi\rangle_S$ according to

$$|\psi\rangle_S \mapsto \sum_{\boldsymbol{E}_a \in \mathcal{E}} \boldsymbol{E}_a |\psi\rangle_S \otimes |\varphi_a\rangle_E = |\psi\rangle_S \otimes \left( \sum_{\boldsymbol{E}_a \in \mathcal{E}} C_a |\varphi_a\rangle_E \right) + |\text{orthog}\rangle_{SE}, \tag{7.48}$$

where $\mathcal{E}$ is the set of all Pauli operators of weight up to $t$, and $|\text{orthog}\rangle$ denotes a vector orthogonal to the code subspace.

Now we can perform an orthogonal measurement on the system $S$, with two possible outcomes: the state is projected onto either the code subspace $\mathcal{H}_{\text{code}}$ or the complementary subspace of $\mathcal{H}_S$. If the first (good) outcome is obtained, the undamaged state $|\psi\rangle$ is recovered. If the second outcome (bad) is found, an error has been detected. We conclude that our QECC with distance $d$ can detect $d{-}1$ errors. Thus a QECC that corrects $t$ errors can detect $2t$ errors.

### 7.4.4 Quantum codes and entanglement

A QECC protects quantum information from error by encoding it *nonlocally*, that is, by sharing it among many qubits in a block. Thus a quantum codeword is a highly entangled state.

In fact, a distance $d = t{+}1$ *nondegenerate* code has the following property: Choose any state $|\psi\rangle$ in the code subspace and any $t$ qubits in the block. Trace over the remaining $n{-}t$ qubits to obtain

$$\boldsymbol{\rho}_{(t)} = \text{tr}_{(n-t)} |\psi\rangle\langle\psi|, \tag{7.49}$$

the density operator of the $t$ qubits. Then this density operator is maximally mixed:

$$\boldsymbol{\rho}_{(t)} = \frac{1}{2^t}\boldsymbol{I}. \tag{7.50}$$

In any distance-($t$+1) code, whether degenerate or not, we cannot acquire any information about the encoded data by observing any $t$ qubits in the block; that is, $\boldsymbol{\rho}_{(t)}$ is a constant, independent of the codeword. But our conclusion that the density operator of any $t$ qubits must be maximally mixed applies only if the code is nondegenerate.

To verify the property eq.(7.50), we note that for a nondegenerate distance-($t$+1) code,

$$\boldsymbol{\Pi}\boldsymbol{E}_a\boldsymbol{\Pi} = 0 \tag{7.51}$$

for any $\boldsymbol{E}_a$ of nonzero weight up to $t$, so that

$$\mathrm{tr}(\boldsymbol{\rho}_{(t)}\boldsymbol{E}_a) = \langle\psi|\boldsymbol{E}_a|\psi\rangle = \langle\psi|\boldsymbol{\Pi}\boldsymbol{E}_a\boldsymbol{\Pi}|\psi\rangle = 0, \tag{7.52}$$

for any $t$-qubit Pauli operator $\boldsymbol{E}_a$ other than the identity. Now $\boldsymbol{\rho}_{(t)}$, like any Hermitian $2^t \times 2^t$ matrix, can be expanded in terms of Pauli operators:

$$\boldsymbol{\rho}_{(t)} = \left(\frac{1}{2^t}\right)\boldsymbol{I} + \sum_{\boldsymbol{E}_b \neq \boldsymbol{I}} \rho_b \boldsymbol{E}_b. \tag{7.53}$$

Since the Pauli operators satisfy

$$\left(\frac{1}{2^t}\right)\mathrm{tr}(\boldsymbol{E}_b^\dagger \boldsymbol{E}_a) = \delta_{ba}, \tag{7.54}$$

we find that each $\rho_b = 0$, and we conclude that $\boldsymbol{\rho}_{(t)}$ is a multiple of the identity. That is, for any state in the code space, and any set of $t$ qubits in the code block, these qubits are maximally entangled with the complementary set of qubits in the block.

## 7.5 Failure probability

### *7.5.1 Fidelity bound*

If the support of the error map contains only the Pauli operators in the set $\mathcal{E}$ that we know how to correct, then we can recover the encoded quantum information with perfect fidelity. But in a realistic error model, there will be a small but nonzero amplitude for errors that are not in $\mathcal{E}$, so that the recovered state will not be perfect. What can we say about the fidelity of the recovered state?

The Pauli operator expansion of the error map can be divided into a sum over the "good" operators (those in $\mathcal{E}$), and the "bad" ones (those not in $\mathcal{E}$), so that it acts on a state $|\psi\rangle$ in the code subspace according to

$$\begin{aligned} |\psi\rangle &\mapsto \sum_a \boldsymbol{E}_a|\psi\rangle \otimes |\varphi_a\rangle_E \\ &= \sum_{\boldsymbol{E}_a \in \mathcal{E}} \boldsymbol{E}_a|\psi\rangle \otimes |\varphi_a\rangle_E + \sum_{\boldsymbol{E}_b \notin \mathcal{E}} \boldsymbol{E}_b|\psi\rangle \otimes |\varphi_b\rangle_E \\ &:= |\mathrm{GOOD}\rangle + |\mathrm{BAD}\rangle. \end{aligned} \tag{7.55}$$

The recovery operation then maps $|\mathrm{GOOD}\rangle$ to a state $|\mathrm{GOOD}'\rangle$ of data, environment, and ancilla, and $|\mathrm{BAD}\rangle$ to a state $|\mathrm{BAD}'\rangle$, so that after recovery we obtain the state

$$|\mathrm{GOOD}'\rangle + |\mathrm{BAD}'\rangle; \tag{7.56}$$

here (since recovery works perfectly acting on the good state)

$$|\mathrm{GOOD}'\rangle = |\psi\rangle \otimes |\chi\rangle_{EA}, \tag{7.57}$$

where $|\chi\rangle_{EA}$ is some (unnormalized) state of the environment and ancilla.

Suppose that the states $|\mathrm{GOOD}\rangle$ and $|\mathrm{BAD}\rangle$ are orthogonal. This would hold if, for example, all of the "good" states of the environment are orthogonal to all of the "bad" states; that is, if

$$\langle\varphi_a|\varphi_b\rangle = 0 \quad \text{for} \quad \boldsymbol{E}_a \in \mathcal{E},\ \boldsymbol{E}_b \notin \mathcal{E}. \tag{7.58}$$

Let $\boldsymbol{\rho}'$ denote the density matrix of the recovered state, obtained by tracing out the environment and ancilla, and let

$$F = \langle\psi|\boldsymbol{\rho}'|\psi\rangle \tag{7.59}$$

be its fidelity with the initial code state. Because the recovery map is an isometry, the states $|\mathrm{BAD}'\rangle$ and $|\mathrm{GOOD}'\rangle$ are also orthogonal (that is, $|\mathrm{BAD}'\rangle$ has no component along $|\psi\rangle \otimes |\chi\rangle_{EA}$). Therefore the fidelity is

$$F = \langle\psi|\boldsymbol{\rho}_{\mathrm{GOOD}'}|\psi\rangle + \langle\psi|\boldsymbol{\rho}_{\mathrm{BAD}'}|\psi\rangle. \tag{7.60}$$

where

$$\boldsymbol{\rho}_{\mathrm{GOOD}'} = \mathrm{tr}_{EA}\left(|\mathrm{GOOD}'\rangle\langle\mathrm{GOOD}'|\right). \quad \boldsymbol{\rho}_{\mathrm{BAD}'} = \mathrm{tr}_{EA}\left(|\mathrm{BAD}'\rangle\langle\mathrm{BAD}'|\right), \tag{7.61}$$

and hence

$$F \geq \ \langle\psi|\boldsymbol{\rho}_{\mathrm{GOOD}'}|\psi\rangle = \| \ |\chi\rangle_{EA} \ \|^2 = \| \ |\mathrm{GOOD}'\rangle \ \|^2 \ . \tag{7.62}$$

Furthermore, since the recovery operation is an isometry, we have $\| \ |\mathrm{GOOD}'\rangle \ \| = \| \ |\mathrm{GOOD}\rangle \ \|$, and hence

$$F \geq \ \| \ |\mathrm{GOOD}\rangle \ \|^2 = \left\| \sum_{\boldsymbol{E}_a \in \mathcal{E}} \boldsymbol{E}_a |\psi\rangle \otimes |\varphi_a\rangle_E \right\|^2 . \tag{7.63}$$

In general, though, $|\mathrm{BAD}\rangle$ need not be orthogonal to $|\mathrm{GOOD}\rangle$, so that $|\mathrm{BAD}'\rangle$ need not be orthogonal to $|\mathrm{GOOD}'\rangle$. Then $|\mathrm{BAD}'\rangle$ might have a component along $|\mathrm{GOOD}'\rangle$ that interferes destructively with $|\mathrm{GOOD}'\rangle$ and so reduces the fidelity. We can still obtain a lower bound on the fidelity in this more general case by resolving $|\mathrm{BAD}'\rangle$ into a component along $|\mathrm{GOOD}'\rangle$ and an orthogonal component, as

$$|\mathrm{BAD}'\rangle = |\mathrm{BAD}'_{\|}\rangle + |\mathrm{BAD}'_{\perp}\rangle. \tag{7.64}$$

Then reasoning just as above we obtain

$$F \geq \ \| \ |\mathrm{GOOD}'\rangle + |\mathrm{BAD}'_{\|}\rangle \ \|^2 \ . \tag{7.65}$$

Of course, since both the error operation and the recovery operation are isometries, the complete state $|\mathrm{GOOD}'\rangle + |\mathrm{BAD}'\rangle$ is normalized, or

$$\| \ |\mathrm{GOOD}'\rangle + |\mathrm{BAD}'_{\|}\rangle \ \|^2 + \| \ |\mathrm{BAD}'_{\perp}\rangle \ \|^2 = 1, \tag{7.66}$$

and eq. (7.65) becomes

$$F \geq \ 1 - \| \ |\mathrm{BAD}'_{\perp}\rangle \ \|^2 \ . \tag{7.67}$$

Finally, the norm of $|\mathrm{BAD}'_\perp\rangle$ cannot exceed the norm of $|\mathrm{BAD}'\rangle$, so we conclude that

$$1 - F \leq \quad \| \, |\mathrm{BAD}'\rangle \, \|^2 = \| |\mathrm{BAD}\rangle \|^2 \equiv \left\| \sum_{\boldsymbol{E}_b \notin \mathcal{E}} \boldsymbol{E}_b |\psi\rangle \otimes |\varphi_b\rangle_E \right\|^2 . \tag{7.68}$$

This is our general bound on the "failure probability" of the recovery operation. The result eq.(7.63) can be recovered from eq.(7.68) in the special case where $|\mathrm{GOOD}\rangle$ and $|\mathrm{BAD}\rangle$ are orthogonal states.

### 7.5.2 Uncorrelated noise

Let's now consider some implications of these results for the case where errors acting on distinct qubits are completely uncorrelated. In that case, the error channel is a tensor product of single-qubit channels. If the noise acts on all the qubits in the same way, then the noise acting on the $n$-qubits in the code block is $\mathcal{N}^{\otimes n}$, where the single-qubit channel $\mathcal{N}$ has the dilation

$$\begin{aligned} |\psi\rangle \mapsto |\psi\rangle \otimes |\varphi_I\rangle_E + \boldsymbol{X}|\psi\rangle \otimes |\varphi_X\rangle_E + \boldsymbol{Y}|\psi\rangle \otimes |\varphi_Y\rangle_E \\ + \boldsymbol{Z}|\psi\rangle \otimes |\varphi_Z\rangle_E . \end{aligned} \tag{7.69}$$

The effect of the noise on encoded information is especially easy to analyze if we suppose further that each of the three states of the environment $|\varphi_{X,Y,Z}\rangle$ is orthogonal to the state $|\varphi_I\rangle$. In that case, a record of whether or not an error occurred for each qubit is permanently imprinted on the environment, and it is sensible to speak of a *probability* of error $\varepsilon$ for each qubit, where

$$\langle \varphi_I | \varphi_I \rangle = 1 - \varepsilon . \tag{7.70}$$

If our quantum code can correct $t$ errors, then the "good" Pauli operators have weight up to $t$, and the "bad" Pauli operators have weight greater than $t$; recovery is certain to succeed unless at least $t+1$ qubits are subjected to errors. It follows that the fidelity obeys the bound

$$1 - F \;\leq\; \sum_{s=t+1}^{n} \binom{n}{s} \varepsilon^s \, (1-\varepsilon)^{n-s} \;\leq\; \binom{n}{t+1} \varepsilon^{t+1} . \tag{7.71}$$

(For each of the $\binom{n}{t+1}$ ways of choosing $t+1$ locations, the probability that errors occur at every one of those locations is $\varepsilon^{t+1}$, where we disregard whether additional errors occur at the remaining $n-t-1$ locations. Therefore, the final expression in eq. (7.71) is an upper bound on the probability that at least $t+1$ errors occur in the block of $n$ qubits.) For $\varepsilon$ small and $t$ large, the fidelity of the encoded data is a substantial improvement over the fidelity $F = 1 - O(\varepsilon)$ maintained by an unprotected qubit.

For a general noise map acting on a single qubit, there is no clear notion of an "error probability;" the state of the qubit and its environment obtained when the Pauli operator $\boldsymbol{I}$ acts is not orthogonal to (and so cannot be perfectly distinguished from) the state obtained when the Pauli operators $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ act. In the extreme case where there is no entanglement with the environment at all, the errors arise because unknown unitary transformations act on the qubits.

Consider uncorrelated unitary errors acting on the $n$ qubits in the code block, each of the form (up to an irrelevant phase)

$$\boldsymbol{U}^{(1)} = \sqrt{1-\varepsilon}\boldsymbol{I} + i\sqrt{\varepsilon}\,\boldsymbol{W}, \tag{7.72}$$

where $\boldsymbol{W}$ is a (traceless, Hermitian) linear combination of $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$, satisfying $\boldsymbol{W}^2 = \boldsymbol{I}$. If the state $|\psi\rangle$ of the qubit is prepared, and then the unitary error eq. (7.72) occurs, the fidelity of the resulting damaged state with the initial state is

$$F = \left|\langle\psi|\boldsymbol{U}^{(1)}|\psi\rangle\right|^2 = 1 - \varepsilon\left(1 - (\langle\psi|\boldsymbol{W}|\psi\rangle)^2\right) \;\geq\; 1-\varepsilon. \tag{7.73}$$

If a unitary error of the form eq. (7.72) acts on each of the $n$ qubits in the code block, and the resulting state is expanded in terms of Pauli operators as in eq. (7.55), then the state $|\mathrm{BAD}\rangle$ (which arises from terms in which $\boldsymbol{W}$ acts on at least $t+1$ qubits) has a norm of order $(\sqrt{\varepsilon})^{t+1}$, and eq. (7.68) becomes

$$1 - F = O(\varepsilon^{t+1}). \tag{7.74}$$

We see that coding provides an improvement in fidelity of the same order irrespective of whether the uncorrelated errors are due to decoherence or due to unknown unitary transformations.

To avoid confusion, let us emphasize the meaning of "uncorrelated" for the purpose of the above discussion. We consider a unitary error acting on $n$ qubits to be "uncorrelated" if it is a tensor product of single-qubit unitary transformations, irrespective of how the unitaries acting on distinct qubits might be related to one another. For example, an "error" whereby all qubits rotate by the same angle $\theta \ll 1$ about a common axis is effectively dealt with by quantum error correction; after recovery the fidelity will be $F = 1 - O(\theta^{2(t+1)})$, if the code can protect against $t$ errors. In contrast, an $n$-qubit unitary error that would cause more trouble is one of the form $\boldsymbol{U}^{(n)} \sim \boldsymbol{I} + i\theta\boldsymbol{E}^{(n)}_{\mathrm{bad}}$, where $\boldsymbol{E}^{(n)}_{\mathrm{bad}}$ is an $n$-qubit Pauli operator whose weight is greater than $t$. Then $|\mathrm{BAD}\rangle$ has a norm of order $\theta$, and the typical fidelity after recovery will be $F = 1 - O(\theta^2)$.

We should also note another important difference between unitary errors and decoherence errors. As discussed in Chapter 3, when decoherence is described by a Markovian master equation, errors are due to "quantum jumps" which occur at a constant rate $\Gamma$ per unit time, and the probability $\varepsilon = \Gamma\tau$ that an error occurs during a time interval of length $\tau$ increases linearly with $\tau$. In contrast, unitary errors due to imperfect control of the system Hamiltonian cause the quantum state to rotate by an angle $\theta$ which accumulates linearly in $\tau$, so that the corresponding infidelity $\sim \theta^2$ increases quadratically with $\tau$. We'll return to this distinction when we discuss fault-tolerant quantum computing in Chapter 8.

## 7.6 Classical Linear Codes

Quantum error-correcting codes were first invented in the mid-1990s, but classical error-correcting codes have a much longer history. Beginning in the 1940s, a remarkably beautiful and powerful theory of classical coding has been erected. Much of this theory can be exploited in the construction of QECC's. Here we will quickly review just a few elements of the classical theory, confining our attention to binary linear codes.

In a binary code, $k$ bits are encoded in a binary string of length $n$. That is, from

among the $2^n$ strings of length $n$, we designate a subset containing $2^k$ strings – the codewords. A $k$-bit message is encoded by selecting one of these $2^k$ codewords.

In the special case of a binary linear code, the codewords form a $k$-dimensional closed linear subspace $C$ of the binary vector space $\mathbb{F}_2^n$. That is, the bitwise XOR of two codewords is another codeword. The space $C$ of the code is spanned by a basis of $k$ vectors $v_1, v_2, \ldots, v_k$; an arbitrary codeword may be expressed as a linear combination of these basis vectors:

$$v(\alpha_1, \ldots, \alpha_k) = \sum_i \alpha_i v_i \ , \tag{7.75}$$

where each $\alpha_i \in \{0, 1\}$, and addition is modulo 2. We may say that the length-$n$ vector $v(\alpha_1 \ldots \alpha_k)$ encodes the $k$-bit message $\alpha = (\alpha_1, \ldots, \alpha_k)$.

The $k$ basis vectors $v_1, \ldots v_k$ may be assembled into a $k \times n$ matrix

$$G = \begin{pmatrix} v_1 \\ \vdots \\ v_k \end{pmatrix} , \tag{7.76}$$

called the *generator matrix* of the code. Then in matrix notation, eq. (7.75) can be rewritten as

$$v(\alpha) = \alpha G \ ; \tag{7.77}$$

the matrix $G$, acting to the left, encodes the message $\alpha$.

An alternative way to characterize the $k$-dimensional code subspace of $F_2^n$ is to specify $n - k$ linear constraints. There is an $(n-k) \times n$ matrix $H$ such that

$$Hv = 0 \tag{7.78}$$

for all those and only those vectors $v$ in the code $C$. This matrix $H$ is called the parity check matrix of the code $C$. The rows of $H$ are $n - k$ linearly independent vectors, and the code space is the space of vectors that are *orthogonal* to all of these vectors. Orthogonality is defined with respect to the mod 2 bitwise inner product; two length-$n$ binary strings are orthogonal is they "collide" (both take the value 1) at an even number of locations. Note that

$$HG^T = 0 \ ; \tag{7.79}$$

where $G^T$ is the transpose of $G$; the rows of $G$ are orthogonal to the rows of $H$.

For a classical bit, the only kind of error is a bit flip. An error occurring in an $n$-bit string can be characterized by an $n$-component vector $e$, where the 1's in $e$ mark the locations where errors occur. When afflicted by the error $e$, the string $v$ becomes

$$v \to v + e \ . \tag{7.80}$$

Errors can be detected by applying the parity check matrix. If $v$ is a codeword, then

$$H(v + e) = Hv + He = He \ . \tag{7.81}$$

$He$ is called the syndrome of the error $e$. Denote by $\mathcal{E}$ the set of errors $\{e_i\}$ that we wish to be able to correct. Error recovery will be possible if and only if all errors $e_i$ have distinct syndromes. If this is the case, we can unambiguously diagnose the error given the syndrome $He$, and we may then recover by flipping the bits specified by $e$ as in

$$v + e \to (v + e) + e = v \ . \tag{7.82}$$

On the other hand, if $He_1 = He_2$ for $e_1 \neq e_2$ then we may misinterpret an $e_1$ error as an $e_2$ error; our attempt at recovery then has the effect

$$v + e_1 \to v + (e_1 + e_2) \neq v. \tag{7.83}$$

The recovered message $v + e_1 + e_2$ lies in the code, but it differs from the intended message $v$; the encoded information has been damaged.

The *distance* $d$ of a code $C$ is the minimum weight of any vector $v \in C$, where the *weight* is the number of 1's in the string $v$. A linear code with distance $d = 2t + 1$ can correct $t$ errors; the code assigns a distinct syndrome to each $e \in \mathcal{E}$, where $\mathcal{E}$ contains all vectors of weight $t$ or less. This is so because, if $He_1 = He_2$, then

$$0 = He_1 + He_2 = H(e_1 + e_2) \ , \tag{7.84}$$

and therefore $e_1 + e_2 \in C$. But if $e_1$ and $e_2$ are unequal and each has weight no larger than $t$, then the weight of $e_1 + e_2$ is greater than zero and no larger than $2t$. Since $d = 2t + 1$, there is no such vector in $C$. Hence $He_1$ and $He_2$ cannot be equal.

A useful concept in classical coding theory is that of the *dual code*. We have seen that the $k \times n$ generator matrix $G$ and the $(n - k) \times n$ parity check matrix $H$ of a code $C$ are related by $HG^T = 0$. Taking the transpose, it follows that $GH^T = 0$. Thus we may regard $H^T$ as the generator and $G$ as the parity check of an $(n - k)$-dimensional code, which is denoted $C^\perp$ and called the dual of $C$. In other words, $C^\perp$ is the orthogonal complement of $C$ in $F_2^n$. A vector is self-orthogonal if it has even weight, so it is possible for $C$ and $C^\perp$ to intersect. A code contains its dual if all of its codewords have even weight and are mutually orthogonal. If $n = 2k$ it is possible that $C = C^\perp$, in which case $C$ is said to be self-dual.

An identity relating the code $C$ and its dual $C^\perp$ will prove useful in the following section:

$$\sum_{v \in C} (-1)^{v \cdot u} = \begin{cases} 2^k & u \in C^\perp \\ 0 & u \notin C^\perp \end{cases} . \tag{7.85}$$

The nontrivial content of the identity is the statement that the sum vanishes for $u \notin C^\perp$. This readily follows from the familiar identity

$$\sum_{v \in \{0,1\}^k} (-1)^{v \cdot w} = 0, w \neq 0, \tag{7.86}$$

where $v$ and $w$ are strings of length $k$. We can express $v \in G$ as

$$v = \alpha G, \tag{7.87}$$

where $\alpha$ is a $k$-vector. Then

$$\sum_{v \in C} (-1)^{v \cdot u} = \sum_{\alpha \in \{0,1\}^k} (-1)^{\alpha \cdot Gu} = 0, \tag{7.88}$$

for $Gu \neq 0$. Since G, the generator matrix of $C$, is the parity check matrix for $C^\perp$, we conclude that the sum vanishes for $u \notin C^\perp$.

## 7.7 CSS Codes

Principles from the theory of classical linear codes can be adapted to the construction of quantum error-correcting codes. We will describe here a family of QECC's, the Calderbank–Shor–Steane (or CSS) codes, that exploit the concept of a dual code.

Let $C_1$ be a classical linear code with $(n-k_1)\times n$ parity check matrix $H_1$, and let $C_2$ be a *subcode* of $C_1$, with $(n-k_2)\times n$ parity check $H_2$, where $k_2 < k_1$. The first $n-k_1$ rows of $H_2$ coincide with those of $H_1$, but there are $k_1-k_2$ additional linearly independent rows; thus each word in $C_2$ is contained in $C_1$, but the words in $C_2$ also obey some additional linear constraints.

The subcode $C_2$ defines an equivalence relation in $C_1$; we say that $u, v \in C_1$ are equivalent ($u \equiv v$) if and only if there is a $w$ in $C_2$ such that $u = v + w$. The equivalence classes are the *cosets* of $C_2$ in $C_1$.

A $CSS$ code is a $k = k_1 - k_2$ quantum code that associates a codeword with each equivalence class. Each element of a basis for the code subspace can be expressed as

$$|\bar{w}\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{v\in C_2} |v+w\rangle \ , \tag{7.89}$$

an equally weighted superposition of all the words in the coset represented by $w$. There are $2^{k_1-k_2}$ cosets, and hence $2^{k_1-k_2}$ linearly independent codewords. The states $|\bar{w}\rangle$ are evidently normalized and mutually orthogonal; that is, $\langle\bar{w}|\bar{w}'\rangle = 0$ if $w$ and $w'$ belong to different cosets.

Now consider what happens to the codeword $|\bar{w}\rangle$ if we apply the bitwise Hadamard transform $\boldsymbol{H}^{(n)}$:

$$\begin{aligned} \boldsymbol{H}^{(n)} : |\bar{w}\rangle_F &\equiv \frac{1}{\sqrt{2^{k_2}}} \sum_{v\in C_2} |v+w\rangle \\ \to |\bar{w}\rangle_P &\equiv \frac{1}{\sqrt{2^n}} \sum_u \frac{1}{\sqrt{2^{k_2}}} \sum_{v\in C_2} (-1)^{u\cdot v}(-1)^{u\cdot w}|u\rangle \\ &= \frac{1}{\sqrt{2^{n-k_2}}} \sum_{u\in C_2^\perp} (-1)^{u\cdot w}|u\rangle \ ; \end{aligned} \tag{7.90}$$

we obtain a coherent superposition, weighted by phases, of words in the dual code $C_2^\perp$ (in the last step we have used the identity eq. (7.85)). It is again manifest in this last expression that the codeword depends only on the $C_2$ coset that $w$ represents — shifting $w$ by an element of $C_2$ has no effect on $(-1)^{u\cdot w}$ if $u$ is in the code dual to $C_2$.

Now suppose that the code $C_1$ has distance $d_1$ and the code $C_2^\perp$ has distance $d_2^\perp$, such that

$$\begin{aligned} d_1 &\geq 2t_F + 1 \ , \\ d_2^\perp &\geq 2t_P + 1 \ . \end{aligned} \tag{7.91}$$

Then we can see that the corresponding CSS code can correct $t_F$ bit flips and $t_P$ phase flips. If $e$ is a binary string of length $n$, let $\boldsymbol{E}_e^{(\mathrm{flip})}$ denote the Pauli operator with an $\boldsymbol{X}$ acting at each location $i$ where $e_i = 1$; it acts on the state $|v\rangle$ according to

$$\boldsymbol{E}_e^{(\mathrm{flip})} : |v\rangle \to |v+e\rangle \ . \tag{7.92}$$

And let $\boldsymbol{E}_e^{(\mathrm{phase})}$ denote the Pauli operator with a $\boldsymbol{Z}$ acting where $e_i = 1$; its action is

$$\boldsymbol{E}_e^{(\mathrm{phase})} : |v\rangle \to (-1)^{v.e}|v\rangle \ , \tag{7.93}$$

which in the Hadamard rotated basis becomes

$$\boldsymbol{E}_e^{(\text{phase})} : |u\rangle \to |u+e\rangle \ . \tag{7.94}$$

Now, in the original basis (the $F$ or "flip" basis), each basis state $|\bar{w}\rangle_F$ of the CSS code is a superposition of words in the code $C_1$. To diagnose bit flip error, we perform on data and ancilla the unitary transformation

$$|v\rangle \otimes |0\rangle_A \to |v\rangle \otimes |H_1 v\rangle_A \ , \tag{7.95}$$

and then measure the ancilla. The measurement result $H_1 e_F$ is the *bit flip syndrome*. If the number of flips is $t_F$ or fewer, we may correctly infer from this syndrome that bit flips have occurred at the locations labeled by $e_F$. We recover by applying $\boldsymbol{X}$ to the qubits at those locations.

To correct phase errors, we first perform the bitwise Hadamard transformation to rotate from the $F$ basis to the $P$ ("phase") basis. In the $P$ basis, each basis state $|\bar{w}\rangle_P$ of the CSS code is a superposition of words in the code $C_2^\perp$. To diagnose phase errors, we perform a unitary transformation

$$|v\rangle \otimes |0\rangle_A \to |v\rangle \otimes |G_2 v\rangle_A \ , \tag{7.96}$$

and measure the ancilla ($G_2$, the generator matrix of $C_2$, is also the parity check matrix of $C_2^\perp$). The measurement result $G_2 e_P$ is the *phase error syndrome*. If the number of phase errors is $t_P$ or fewer, we may correctly infer from this syndrome that phase errors have occurred at locations labeled by $e_P$. We recover by applying $\boldsymbol{X}$ (in the $P$ basis) to the qubits at those locations. Finally, we apply the bitwise Hadamard transformation once more to rotate the codewords back to the original basis. (Equivalently, we may recover from the phase errors by applying $\boldsymbol{Z}$ to the affected qubits after the rotation back to the $F$ basis.)

If $e_F$ has weight less than $d_1$ and $e_P$ has weight less than $d_2^\perp$, then

$$\langle \bar{w} | \boldsymbol{E}_{e_P}^{(\text{phase})} \boldsymbol{E}_{e_F}^{(\text{flip})} | \bar{w}' \rangle = 0 \tag{7.97}$$

(unless $e_F = e_P = 0$). Any Pauli operator can be expressed as a product of a phase operator and a flip operator — a $\boldsymbol{Y}$ error is merely a bit flip and phase error both afflicting the same qubit. So the distance $d$ of a CSS code satisfies

$$d \ \geq \ \min(d_1, d_2^\perp) \ . \tag{7.98}$$

CSS codes have the special property (not shared by more general QECC's) that the recovery procedure can be divided into two separate operations, one to correct the bit flips and the other to correct the phase errors.

The unitary transformation eq. (7.95) (or eq. (7.96)) can be implemented by executing a simple quantum circuit. Associated with each of the $n - k_1$ rows of the parity check matrix $H_1$ is a bit of the syndrome to be extracted. To find the $a$th bit of the syndrome, we prepare an ancilla bit in the state $|0\rangle_{A,a}$, and for each value of $\lambda$ with $(H_1)_{a\lambda} = 1$, we execute a controlled-NOT gate with the ancilla bit as the target and qubit $\lambda$ in the data block as the control. When measured, the ancilla qubit reveals the value of the parity check bit $\sum_\lambda (H_1)_{a\lambda} v_\lambda$.

Schematically, the full error correction circuit for a CSS code has the form:

– Figure –

Separate syndromes are measured to diagnose the bit flip errors and the phase errors. An important special case of the CSS construction arises when a code $C$ contains its dual $C^\perp$. Then we may choose $C_1 = C$ and $C_2 = C^\perp \subseteq C$; the $C$ parity check is computed in both the $F$ basis and the $P$ basis to determine the two syndromes.

## 7.8 The 7-Qubit Code

The simplest of the CSS codes is the $[[n, k, d]] = [7, 1, 3]$ quantum code first formulated by Andrew Steane. It is constructed from the classical 7-bit Hamming code.

The Hamming code is an $[n, k, d] = [7, 4, 3]$ classical code with the $3 \times 7$ parity check matrix

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{7.99}$$

To see that the distance of the code is $d = 3$, first note that the weight-3 string (1110000) passes the parity check and is, therefore, in the code. Now we need to show that there are no vectors of weight 1 or 2 in the code. If $e_1$ has weight 1, then $He_1$ is one of the columns of $H$. But no column of $H$ is trivial (all zeros), so $e_1$ cannot be in the code. Any vector of weight 2 can be expressed as $e_1 + e_2$, where $e_1$ and $e_2$ are distinct vectors of weight 1. But

$$H(e_1 + e_2) = He_1 + He_2 \neq 0, \tag{7.100}$$

because all columns of $H$ are distinct. Therefore $e_1 + e_2$ cannot be in the code.

The rows of $H$ themselves pass the parity check, and so are also in the code. (Contrary to one's usual linear algebra intuition, a nonzero vector over the finite field $F_2$ can be orthogonal to itself.) The generator matrix $G$ of the Hamming code can be written as

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}; \tag{7.101}$$

the first three rows coincide with the rows of $H$, and the weight-3 codeword (1110000) is appended as the fourth row.

The dual of the Hamming code is the $[7, 3, 4]$ code generated by $H$. In this case the dual of the code is actually contained in the code — in fact, it is the *even subcode* of the Hamming code, containing all those and only those Hamming codewords that have even weight. The odd codeword (1110000) is a representative of the nontrivial coset of the even subcode. For the CSS construction, we will choose $C_1$ to be the Hamming code, and $C_2$ to be its dual, the even subcode.. Therefore, $C_2^\perp = C_1$ is again the Hamming code; we will use the Hamming parity check both to detect bit flips in the $F$ basis and to detect phase flips in the $P$ basis.

In the $F$ basis, the two orthonormal codewords of this CSS code, each associated with a distinct coset of the even subcode, can be expressed as

$$|\bar{0}\rangle_F = \frac{1}{\sqrt{8}} \sum_{\substack{\text{even } v \\ \in \text{ Hamming}}} |v\rangle \ ,$$

$$|\bar{1}\rangle_F = \frac{1}{\sqrt{8}} \sum_{\substack{\text{odd } v \\ \in \text{ Hamming}}} |v\rangle \ . \tag{7.102}$$

Since both $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are superpositions of Hamming codewords, bit flips can be diagnosed in this basis by performing an $H$ parity check. In the Hadamard rotated basis, these codewords become

$$\begin{aligned} \boldsymbol{H}^{(7)} : |\bar{0}\rangle_F \to |\bar{0}\rangle_P &\equiv \left(\frac{1}{4}\right) \sum_{v \in \text{ Hamming}} |v\rangle = \frac{1}{\sqrt{2}}(|\bar{0}\rangle_F + |\bar{1}\rangle_F) \\ |\bar{1}\rangle_F \to |\bar{1}\rangle_P &\equiv \left(\frac{1}{4}\right) \sum_{v \in \text{ Hamming}} (-1)^{\text{wt}(v)}|v\rangle = \frac{1}{\sqrt{2}}(|\bar{0}\rangle_F - |\bar{1}\rangle_F). \end{aligned} \tag{7.103}$$

In this basis as well, the states are superpositions of Hamming codewords, so that bit flips in the $P$ basis (phase flips in the original basis) can again be diagnosed with an $H$ parity check. (We note in passing that for this code, performing the bitwise Hadamard transformation also implements a Hadamard rotation on the encoded data, a point that will be relevant to our discussion of fault-tolerant quantum computation in the next chapter.)

Steane's quantum code can correct a single bit flip and a single phase flip on any one of the seven qubits in the block. But recovery will fail if two different qubits both undergo either bit flips or phase flips. If $e_1$ and $e_2$ are two distinct weight-one strings then $He_1 + He_2$ is a sum of two distinct columns of $H$, and hence a third column of $H$ (all seven of the nontrivial strings of length 3 appear as columns of $H$.) Therefore, there is another weight-one string $e_3$ such that $He_1 + He_2 = He_3$, or

$$H(e_1 + e_2 + e_3) = 0 \ ; \tag{7.104}$$

thus $e_1+e_2+e_3$ is a weight-3 word in the Hamming code. We will interpret the syndrome $He_3$ as an indication that the error $v \to v+e_3$ has arisen, and we will attempt to recover by applying the operation $v \to v+e_3$. Altogether then, the effect of the two bit flip errors and our faulty attempt at recovery will be to add $e_1+e_2+e_3$ (an odd-weight Hamming codeword) to the data, which will induce a flip of the *encoded* qubit

$$|\bar{0}\rangle_F \leftrightarrow |\bar{1}\rangle_F. \tag{7.105}$$

Similarly, two phase flips in the $F$ basis are two bit flips in the $P$ basis, which (after the botched recovery) induce on the encoded qubit

$$|\bar{0}\rangle_P \leftrightarrow |\bar{1}\rangle_P, \tag{7.106}$$

or equivalently

$$\begin{aligned} |\bar{0}\rangle_F &\to |\bar{0}\rangle_F \\ |\bar{1}\rangle_F &\to -|\bar{1}\rangle_F, \end{aligned} \tag{7.107}$$

a phase flip of the encoded qubit in the $F$ basis. If there is one bit flip and one phase flip (either on the same qubit or different qubits) then recovery will be successful.

## 7.9 Some Constraints on Code Parameters

Shor's code protects one encoded qubit from an error in any single one of nine qubits in a block, and Steane's code reduces the block size from nine to seven. Can we do better still?

### 7.9.1 The Quantum Hamming bound

To understand how much better we might do, let's see if we can derive any bounds on the distance $d = 2t+1$ of an $[[n, k, d]]$ quantum code, for given $n$ and $k$. At first, suppose we limit our attention to *nondegenerate* codes, which assign a distinct syndrome to each possible error. On a given qubit, there are three possible linearly independent errors $\boldsymbol{X}, \boldsymbol{Y}$, or $\boldsymbol{Z}$. In a block of $n$ qubits, there are $\binom{n}{j}$ ways to choose $j$ qubits that are affected by errors, and three possible errors for each of these qubits; therefore the total number of possible errors of weight up to $t$ is

$$N(t) = \sum_{j=0}^{t} 3^j \binom{n}{j}. \tag{7.108}$$

If there are $k$ encoded qubits, then there are $2^k$ linearly independent codewords. If all $\boldsymbol{E}_a|\bar{j}\rangle$'s are linearly independent, where $\boldsymbol{E}_a$ is any error of weight up to $t$ and $|\bar{i}\rangle$ is any element of a basis for the codewords, then the dimension $2^n$ of the Hilbert space of $n$ qubits must be large enough to accommodate $N(t) \cdot 2^k$ independent vectors; hence

$$N(t) = \sum_{j=0}^{t} 3^j \binom{n}{j} \leq 2^{n-k}. \tag{7.109}$$

This result is called the quantum Hamming bound. An analogous bound applies to classical block codes, but without the factor of $3^j$, since there is only one type of error (a flip) that can affect a classical bit. We also emphasize that the quantum Hamming bound applies only in the case of nondegenerate coding, while the classical Hamming bound applies in general. However, no degenerate quantum codes that violate the quantum Hamming code have yet been constructed (as of January, 1999).

In the special case of a code with one encoded qubit ($k = 1$) that corrects one error ($t = 1$), the quantum Hamming bound becomes

$$1 + 3n \leq 2^{n-1}, \tag{7.110}$$

which is satisfied for $n \geq 5$. In fact, the case $n = 5$ saturates the inequality ($1+15 = 16$). A nondegenerate $[[5, 1, 3]]$ quantum code, if it exists, is *perfect*: The entire 32-dimensional Hilbert space of the five qubits is needed to accommodate all possible one-qubit errors acting on all codewords — there is no wasted space.

### 7.9.2 The no-cloning bound

We could still wonder, though, if there is a *degenerate* $n = 4$ code that can correct one error. In fact, it is easy to see that no such code can exist. We already know that a code that corrects $t$ errors at arbitrary locations can also be used to correct $2t$ errors at known locations. Suppose that we have a $[[4, 1, 3]]$ quantum code. Then we could

encode a single qubit in the four-qubit block, and split the block into two sub-blocks, each containing two qubits.

– Figure –

If we append $|00\rangle$ to each of those two sub-blocks, then the original block has spawned two offspring, each with two located errors. If we were able to correct the two located errors in each of the offspring, we would obtain two identical copies of the parent block — we would have cloned an unknown quantum state, which is impossible. Therefore, no $[[4, 1, 3]]$ quantum code can exist. We conclude that $n = 5$ is the minimal block size of a quantum code that corrects one error, whether the code is degenerate or not.

The same reasoning shows that an $[[n, k \geq 1, d]]$ code can exist only for

$$n > 2(d - 1) \ . \tag{7.111}$$

### *7.9.3 The quantum Singleton bound*

This result eq. (7.111) can be strengthened to

$$n - k \geq 2(d - 1). \tag{7.112}$$

Eq. (7.112) resembles the Singleton bound on classical code parameters,

$$n - k \geq d - 1, \tag{7.113}$$

and so has been called the "quantum Singleton bound." For a classical *linear* code, the Singleton bound is a near triviality: the code can have distance $d$ only if any $d - 1$ columns of the parity check matrix are linearly independent. Since the columns have length $n - k$, at most $n - k$ columns can be linearly independent; therefore $d - 1$ cannot exceed $n - k$. The Singleton bound also applies to nonlinear codes.

The quantum Singleton bound shows that $n = 5$ is the minimal code length for an $[[n,1,3]]$ quantum code, and that $n = 4$ is the minimal length for an $[[n,2,2]]$ code; we'll soon see that $[[5, 1, 3]]$ and $[[4, 2, 2]]$ codes can be constructed. The bound does not exclude the existence of a $[[3,1,2]]$ code, although no such code exists for qubits. However, the bound also applies to codes constructed from higher dimensional objects, and in fact there is a $[[3,1,2]]$ code in which the fundamental particles are qutrits rather than qubits, as you will show in Exercise ??.

An elegant proof of the quantum Singleton bound can be found based on information theory ideas which we will develop in Chapter 10. In Exercise ?? we prove the following statement: Suppose that the code block can be divided into three parts $ABC$ such that erasure of $A$ can be corrected and erasure of $B$ can also be corrected. Then the number of encoded qubits satisfies $k \leq |C|$, where $|C|$ denotes the number of qubits in $C$. To derive the quantum Singleton bound, choose $A$ and $B$ to be disjoint sets each containing $d - 1$ qubits; hence $A$ and $B$ are both correctable. (This is possible because the no-cloning bound ensures that $n > 2(d - 1)$ for $k \geq 1$.). Choose $C$ to be the complement of $AB$ in the code block. Then we have

$$k \leq |C| = n - |A| - |B| = n - 2(d - 1). \tag{7.114}$$

The quantum Singleton bound then follows.

For most values of $n$ and $k$ the quantum Singleton bound is not tight. Rains has obtained the stronger result that an $[[n, k, 2t+1]]$ code with $k \geq 1$ must satisfy

$$t \leq \left\lfloor \frac{n+1}{6} \right\rfloor, \tag{7.115}$$

where $\lfloor x \rfloor$ "floor $x$" is the greatest integer greater than or equal to $x$. Thus, the minimal length of a $k = 1$ code that can correct $t = 1, 2, 3, 4, 5$ errors is $n = 5, 11, 17, 23, 29$ respectively. Codes with all of these parameters have actually been constructed, except for the $[[23, 1, 9]]$ code.

## 7.10 Stabilizer Codes

### *7.10.1 General formulation*

We will be able to construct a (nondegenerate) $[[5, 1, 3]]$ quantum code, but to do so, we will need a more powerful procedure for constructing quantum codes than the CSS procedure.

Recall that to establish a criterion for when error recovery is possible, we found it quite useful to expand an error superoperator in terms of the $n$-qubit Pauli operators. But up until now we have not exploited the group structure of these operators (a product of Pauli operators is a Pauli operator). In fact, we will see that group theory is a powerful tool for constructing QECC's.

For a single qubit, we will find it more convenient now to choose all of the Pauli operators to be represented by real matrices, so I will now use a notation in which $\boldsymbol{Y}$ denotes the anti-hermitian matrix

$$\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{X} = i\boldsymbol{\sigma}y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \tag{7.116}$$

satisfying $\boldsymbol{Y}^2 = -\boldsymbol{I}$. Then the operators

$$\{\pm\boldsymbol{I}, \pm\boldsymbol{X}, \pm\boldsymbol{Y}, \pm\boldsymbol{Z}\} \equiv \pm\{\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}, \tag{7.117}$$

are the elements of a group of order 8.[1] The $n$-fold tensor products of single-qubit Pauli operators also form a group

$$G_n = \pm\{\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}^{\otimes n}, \tag{7.118}$$

of order $|G_n| = 2^{2n+1}$ (since there are $4^n$ possible tensor products, and another factor of 2 for the $\pm$ sign) we will refer to $G_n$ as the *$n$-qubit Pauli group*. (In fact, we will use the term "Pauli group" both to refer to the abstract group $G_n$, and to its dimension-$2^n$ faithful unitary representation by tensor products of $2 \times 2$ matrices; its only irreducible representation of dimension greater than 1.) Note that $G_n$ has the two element center $Z_2 = \{\pm\boldsymbol{I}^{\otimes n}\}$. If we quotient out its center, we obtain the group $\bar{G}_n \equiv G_n/Z_2$; this group can also be regarded as a binary vector space of dimension $2^{2n}$, a property that we will exploit below.

The ($2^n$-dimensional representation of the) Pauli group $G_n$ evidently has these properties:

[1] It is not the quaternionic group but the *other* non-abelian group of order 8 — the symmetry group of the square. The element $\boldsymbol{Y}$, of order 4, can be regarded as the 90° rotation of the plane, while $\boldsymbol{X}$ and $\boldsymbol{Z}$ are reflections about two orthogonal axes.

**(i)** Each $\boldsymbol{M} \in G_n$ is unitary, $\boldsymbol{M}^{-1} = \boldsymbol{M}^\dagger$.

**(ii)** For each element $\boldsymbol{M} \in G_n$, $\boldsymbol{M}^2 = \pm\boldsymbol{I} \equiv \pm\boldsymbol{I}^{\otimes n}$. Furthermore, $\boldsymbol{M}^2 = \boldsymbol{I}$ if the number of $\boldsymbol{Y}$'s in the tensor product is even, and $\boldsymbol{M}^2 = -\boldsymbol{I}$ if the number of $\boldsymbol{Y}$'s is odd.

**(iii)** If $\boldsymbol{M}^2 = \boldsymbol{I}$, then $\boldsymbol{M}$ is hermitian ($\boldsymbol{M} = \boldsymbol{M}^\dagger$); if $\boldsymbol{M}^2 = -\boldsymbol{I}$, then $\boldsymbol{M}$ is anti-hermitian ($\boldsymbol{M} = -\boldsymbol{M}^\dagger$).

**(iv)** Any two elements $\boldsymbol{M}, \boldsymbol{n} \in G_n$ either commute or anti-commute: $\boldsymbol{M}\boldsymbol{n} = \pm\boldsymbol{n}\boldsymbol{M}$.

We will use the Pauli group to characterize a QECC in the following way: Let $S$ denote an abelian subgroup of the $n$-qubit Pauli group $G_n$. Thus all elements of $S$ acting on $\mathcal{H}_{2^n}$ can be simultaneously diagonalized. Then the *stabilizer code* $\mathcal{H}_S \subseteq \mathcal{H}_2 n$ associated with $S$ is the simultaneous eigenspace with eigenvalue 1 of all elements of $S$. That is,

$$|\psi\rangle \in \mathcal{H}_S \quad \text{iff} \quad \boldsymbol{M}|\psi\rangle = |\psi\rangle \text{ for all } \boldsymbol{M} \in S. \tag{7.119}$$

The group $S$ is called the *stabilizer* of the code, since it preserves all of the codewords.

The group $S$ can be characterized by its generators. These are elements $\{\boldsymbol{M}_i\}$ that are *independent* (no one can be expressed as a product of others) and such that each element of $S$ can be expressed as a product of elements of $\{\boldsymbol{M}_i\}$. If $S$ has $n-k$ generators, we can show that the code space $\mathcal{H}_S$ has dimension $2^k$ — there are $k$ encoded qubits.

To verify this, first note that each $\boldsymbol{M} \in S$ must satisfy $\boldsymbol{M}^2 = \boldsymbol{I}$; if $\boldsymbol{M}^2 = -\boldsymbol{I}$, then $\boldsymbol{M}$ cannot have the eigenvalue $+1$. Furthermore, for each $\boldsymbol{M} \neq \pm\boldsymbol{I}$ in $G_n$ that squares to one, the eigenvalues $+1$ and $-1$ have equal degeneracy. This is because for each $\boldsymbol{M} \neq \pm\boldsymbol{I}$, there is an $\boldsymbol{n} \in G_n$ that anti-commutes with $\boldsymbol{M}$,

$$\boldsymbol{n}\boldsymbol{M} = -\boldsymbol{M}\boldsymbol{n} \ ; \tag{7.120}$$

therefore, $\boldsymbol{M}|\psi\rangle = |\psi\rangle$ if and only if $\boldsymbol{M}(\boldsymbol{n}|\psi\rangle) = -\boldsymbol{n}|\psi\rangle$, and the action of the unitary $\boldsymbol{n}$ establishes a $1-1$ correspondence between the $+1$ eigenstates of $\boldsymbol{M}$ and the $-1$ eigenstates. Hence there are $\frac{1}{2}(2^n) = 2^{n-1}$ mutually orthogonal states that satisfy

$$\boldsymbol{M}_1|\psi\rangle = |\psi\rangle \ , \tag{7.121}$$

where $\boldsymbol{M}_1$ is one of the generators of $S$.

Now let $\boldsymbol{M}_2$ be another element of $G_n$ that commutes with $\boldsymbol{M}_1$ such that $\boldsymbol{M}_2 \neq \pm\boldsymbol{I}, \pm\boldsymbol{M}_1$. We can find an $\boldsymbol{n} \in G_n$ that commutes with $\boldsymbol{M}_1$ but anti-commutes with $\boldsymbol{M}_2$; therefore $\boldsymbol{n}$ preserves the $+1$ eigenspace of $\boldsymbol{M}_1$, but within this space, it interchanges the $+1$ and $-1$ eigenstates of $\boldsymbol{M}_2$. It follows that the space satisfying

$$\boldsymbol{M}_1|\psi\rangle = \boldsymbol{M}_2|\psi\rangle = |\psi\rangle, \tag{7.122}$$

has dimension $2^{n-2}$.

Continuing in this way, we note that if $\boldsymbol{M}_j$ is independent of $\{\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots \boldsymbol{M}_{j-1}\}$, then there is an $\boldsymbol{n}$ that commutes with $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_{j-1}$, but anti-commutes with $\boldsymbol{M}_j$ (we'll discuss in more detail below how such an $\boldsymbol{n}$ can be found). Therefore, restricted to the space with $\boldsymbol{M}_1 = \boldsymbol{M}_2 = \ldots = \boldsymbol{M}_{j-1} = 1, \boldsymbol{M}_j$ has as many $+1$ eigenvectors as $-1$ eigenvectors. So adding another generator always cuts the dimension of the simultaneous eigenspace in half. With $n-k$ generators, the dimension of the remaining space is $2^n\,(1/2)^{n-k} = 2^k$.

The stabilizer language is useful because it provides a simple way to characterize the errors that the code can detect and correct. We may think of the $n-k$ stabilizer generators $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_{n-k}$, as the *check operators* of the code, the collective observables that we measure to diagnose the errors. If the encoded information is undamaged, then

we will find $\boldsymbol{M}_i = 1$ for each of the generators; but if $\boldsymbol{M}_i = -1$ for some $i$, then the data is orthogonal to the code subspace and an error has been detected.

Recall that the error superoperator can be expanded in terms of elements $\boldsymbol{E}_a$ of the Pauli group. $A$ particular $\boldsymbol{E}_a$ either commutes or anti-commutes with a particular stabilizer generator $\boldsymbol{M}$. If $\boldsymbol{E}_a$ and $\boldsymbol{M}$ commute, then

$$\boldsymbol{M}\boldsymbol{E}_a|\psi\rangle = \boldsymbol{E}_a\boldsymbol{M}|\psi\rangle = \boldsymbol{E}_a|\psi\rangle, \tag{7.123}$$

for $|\psi\rangle \in \mathcal{H}_S$, so the error preserves the value $\boldsymbol{M} = 1$. But if $\boldsymbol{E}_a$ and $\boldsymbol{M}$ anti-commute, then

$$\boldsymbol{M}\boldsymbol{E}_a|\psi\rangle = -\boldsymbol{E}_a\boldsymbol{M}|\psi\rangle = -\boldsymbol{E}_a|\psi\rangle, \tag{7.124}$$

so that the error flips the value of $\boldsymbol{M}$, and the error can be detected by measuring $\boldsymbol{M}$.

For stabilizer generators $\boldsymbol{M}_i$ and errors $\boldsymbol{E}_a$, we may write

$$\boldsymbol{M}_i\boldsymbol{E}_a = (-1)^{s_{ia}}\boldsymbol{E}_a\boldsymbol{M}_i. \tag{7.125}$$

The $s_{ia}$'s, $i = 1, \ldots, n-k$ constitute a *syndrome* for the error $\boldsymbol{E}_a$, as $(-1)^{s_{ia}}$ will be the result of measuring $\boldsymbol{M}_i$ if the error $\boldsymbol{E}_a$ occurs. In the case of a nondegenerate code, the $s_{ia}$'s will be distinct for all $\boldsymbol{E}_a \in \mathcal{E}$, so that measuring the $n-k$ stabilizer generators will diagnose the error completely.

More generally, let us find a condition to be satisfied by the stabilizer that is sufficient to ensure that error recovery is possible. Recall that it is sufficient that, for each $\boldsymbol{E}_a, \boldsymbol{E}_b \in \mathcal{E}$, and normalized $|\psi\rangle$ in the code subspace, we have

$$\langle\psi|\boldsymbol{E}_a^\dagger\boldsymbol{E}_b|\psi\rangle = C_{ab}, \tag{7.126}$$

where $C_{ab}$ is independent of $|\psi\rangle$. We can see that this condition is satisfied provided that, for each $\boldsymbol{E}_a, \boldsymbol{E}_b \in \mathcal{E}$, one of the following holds:

**1)** $\boldsymbol{E}_a^\dagger\boldsymbol{E}_b \in S$ ,

**2)** There is an $\boldsymbol{M} \in S$ that anti-commutes with $\boldsymbol{E}_a^\dagger\boldsymbol{E}_b$.

**Proof:** In case (1) $\langle\psi|\boldsymbol{E}_a^\dagger\boldsymbol{E}_b|\psi\rangle = \langle\psi|\psi\rangle = 1$, for $|\psi\rangle \in \mathcal{H}_S$. In case (2), suppose $\boldsymbol{M} \in S$ and $\boldsymbol{M}\boldsymbol{E}_a^\dagger\boldsymbol{E}_b = -\boldsymbol{E}_a^\dagger\boldsymbol{E}_b\boldsymbol{M}$. Then

$$\begin{aligned}\langle\psi|\boldsymbol{E}_a^\dagger\boldsymbol{E}_b|\psi\rangle &= \langle\psi|\boldsymbol{E}_a^\dagger\boldsymbol{E}_b\boldsymbol{M}|\psi\rangle \\ &= -\langle\psi|\boldsymbol{M}\boldsymbol{E}_a^\dagger\boldsymbol{E}_b|\psi\rangle = -\langle\psi|\boldsymbol{E}_a^\dagger\boldsymbol{E}_b|\psi\rangle,\end{aligned} \tag{7.127}$$

and therefore $\langle\psi|\boldsymbol{E}_a^\dagger\boldsymbol{E}_b|\psi\rangle = 0$.

Thus, a *stabilizer code* that corrects $\{\mathcal{E}\}$ is a space $\mathcal{H}_S$ fixed by an abelian subgroup $S$ of the Pauli group, where either (1) or (2) is satisfied by each $\boldsymbol{E}_a^\dagger\boldsymbol{E}_b$ with $\boldsymbol{E}_{a,b} \in \mathcal{E}$. The code is *nondegenerate* if condition (1) is not satisfied for any $\boldsymbol{E}_a^\dagger\boldsymbol{E}_b$.

Evidently we could also just as well choose the code subspace to be any one of the $2^{n-k}$ simultaneous eigenspaces of $n-k$ independent commuting elements of $G_n$. But in fact all of these codes are equivalent. We may regard two stabilizer codes as *equivalent* if they differ only according to how the qubits are labeled, and how the basis for each single-qubit Hilbert space is chosen – that is the stabilizer of one code is transformed to the stabilizer of the other by a permutation of the qubits together with a tensor product of single-qubit transformations. If we partition the stabilizer generators into two sets $\{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_j\}$ and $\{\boldsymbol{M}_{j+1}, \ldots, \boldsymbol{M}_{n-k}\}$, then there exists an $\boldsymbol{n} \in G_n$ that commutes with each member of the first set and anti-commutes with each member of the second

set. Applying $\boldsymbol{n}$ to $|\psi\rangle \in \mathcal{H}_s$ preserves the eigenvalues of the first set while flipping the eigenvalues of the second set. Since $\boldsymbol{n}$ is just a tensor product of single-qubit unitary transformations, there is no loss of generality (up to equivalence) in choosing all of the eigenvalues to be one. Furthermore, since minus signs don't really matter when the stabilizer is specified, we may just as well say that two codes are equivalent if, up to phases, the stabilizers differ by a permutation of the $n$ qubits, and permutations on each individual qubits of the operators $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$.

Recovery may fail if there is an $\boldsymbol{E}_a^\dagger \boldsymbol{E}_b$ that *commutes* with the stabilizer but does not lie in the stabilizer. This is an operator that preserves the code subspace $\mathcal{H}_S$ but may act nontrivially in that space; thus it can modify encoded information. Since $\boldsymbol{E}_a|\psi\rangle$ and $\boldsymbol{E}_b|\psi\rangle$ have the same syndrome, we might mistakenly interpret an $\boldsymbol{E}_a$ error as an $\boldsymbol{E}_b$ error; the effect of the error together with the attempt at recovery is that $\boldsymbol{E}_b^\dagger \boldsymbol{E}_a$ gets applied to the data, which can cause damage.

A stabilizer code with distance $d$ has the property that each $\boldsymbol{E} \in G_n$ of weight less than $d$ either lies in the stabilizer or anti-commutes with some element of the stabilizer. The code is nondegenerate if the stabilizer contains no elements of weight less than $d$. A distance $d = 2t + 1$ code can correct $t$ errors, and a distance $s + 1$ code can detect $s$ errors or correct $s$ errors at known locations.

### *7.10.2 Symplectic Notation*

Properties of stabilizer codes are often best explained and expressed using the language of linear algebra. The stabilizer $S$ of the code, an order $2^{n-k}$ abelian subgroup of the Pauli group with all elements squaring to the identity, can equivalently be regarded as a dimension $n - k$ closed linear subspace of $F_2^{2n}$, self orthogonal with respect to a certain (symplectic) inner product.

The group $\bar{G}_n = G_n/Z_2$ is isomorphic to the binary vector space $F_2^{2n}$. We establish this by observing that, since $\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{X}$, any element $\boldsymbol{M}$ of the Pauli group (up to the $\pm$ sign) can be expressed as a product of $\boldsymbol{Z}$'s and $\boldsymbol{X}$'s; we may write

$$\boldsymbol{M} = \boldsymbol{Z}_M \cdot \boldsymbol{X}_M \tag{7.128}$$

where $\boldsymbol{Z}_M$ is a tensor product of $\boldsymbol{Z}$'s and $\boldsymbol{X}_M$ is a tensor product of $\boldsymbol{X}$'s. More explicitly, a Pauli operator may be written as

$$(\alpha|\beta) \equiv \boldsymbol{Z}(\alpha)\boldsymbol{X}(\beta) = \bigotimes_{i=1}^{n} \boldsymbol{Z}^{\alpha_i} \cdot \bigotimes_{i=1}^{n} \boldsymbol{X}^{\beta_i}, \tag{7.129}$$

where $\alpha$ and $\beta$ are binary strings of length $n$. (Then $\boldsymbol{Y}$ acts at the locations where $\alpha$ and $\beta$ "collide.") Multiplication in $\bar{G}_n$ maps to addition in $F_2^{2n}$:

$$(\alpha|\beta)(\alpha'|\beta') = (-1)^{\alpha' \cdot \beta}(\alpha + \alpha'|\beta + \beta') \ ; \tag{7.130}$$

the phase arises because $\alpha' \cdot \beta$ counts the number of times a $\boldsymbol{Z}$ is interchanged with a $\boldsymbol{X}$ as the product is rearranged into the standard form of eq. (7.129).

It follows from eq. (7.130) that the commutation properties of the Pauli operators can be expressed in the form

$$(\alpha|\beta)(\alpha'|\beta') = (-1)^{\alpha \cdot \beta' + \alpha' \cdot \beta}(\alpha'|\beta')(\alpha|\beta) \tag{7.131}$$

Thus two Pauli operators commute if and only if the corresponding vectors are orthogonal with respect to the "symplectic" inner product

$$\alpha \cdot \beta' + \alpha' \cdot \beta \ . \tag{7.132}$$

We also note that the square of a Pauli operator is

$$(\alpha|\beta)^2 = (-1)^{\alpha\cdot\beta} \boldsymbol{I} \ , \tag{7.133}$$

since $\alpha \cdot \beta$ counts the number of $\boldsymbol{Y}$'s in the operator; it squares to the identity if and only if

$$\alpha \cdot \beta = 0 \ . \tag{7.134}$$

Note that a closed subspace, where each element has this property, is automatically self-orthogonal, since

$$\alpha \cdot \beta' + \alpha' \cdot \beta = (\alpha + \alpha') \cdot (\beta + \beta') - \alpha \cdot \beta - \alpha' \cdot \beta' = 0 \ ; \tag{7.135}$$

in the group language, that is, a subgroup of $G_n$ with each element squaring to $\boldsymbol{I}$ is automatically abelian.

Using the linear algebra language, some of the statements made earlier about the Pauli group can be easily verified by counting linear constraints. Elements are independent if the corresponding vectors are linearly independent over $F_2^{2n}$, so we may think of the $n-k$ generators of the stabilizer as a basis for a linear subspace of dimension $n-k$. We will use the notation $S$ to denote both the linear space and the corresponding abelian group. Then $S^\perp$ denotes the dimension-$n+k$ space of vectors that are orthogonal to each vector in $S$ (with respect to the symplectic inner product). Note that $S^\perp$ contains $S$, since all vectors in $S$ are mutually orthogonal. In the group language, corresponding to $S^\perp$ is the normalizer (or centralizer) group $N(S)$ ($\equiv S^\perp$) of $S$ in $G_n$ — the subgroup of $G_n$ containing all elements that commute with each element of $S$. Since $S$ is abelian, it is contained in its own normalizer, which also contains other elements (to be further discussed below). The stabilizer of a distance $d$ code has the property that each $(\alpha|\beta)$ whose weight $\sum_i(\alpha_i \vee \beta_i)$ is less than $d$ either lies *in* the stabilizer subspace $S$ or lies *outside* the orthogonal space $S^\perp$.

A code can be characterized by its stabilizer, a stabilizer by its generators, and the $n-k$ generators can be represented by an $(n-k) \times 2n$ matrix

$$H = (H_Z|H_X). \tag{7.136}$$

Here each row is a Pauli operator, expressed in the $(\alpha|\beta)$ notation. The syndrome of an error $\boldsymbol{E}_a = (\alpha_a|\beta_a)$ is determined by its commutation properties with the generators $\boldsymbol{M}_i = (\alpha_i'|\beta_i')$; that is

$$s_{ia} = (\alpha_a|\beta_a) \cdot (\alpha_i'|\beta_i') = \alpha_a \cdot \beta_i' + \alpha_i' \cdot \beta_a. \tag{7.137}$$

In the case of a nondegenerate code, each error has a distinct syndrome. If the code is degenerate, there may be several errors with the same syndrome, but we may apply any one of the $\boldsymbol{E}_a^\dagger$ corresponding to the observed syndrome in order to recover.

### *7.10.3 Some examples of stabilizer codes*

**(a) The nine-qubit code**. This $[[9, 1, 3]]$ code has eight stabilizer generators that can

be expressed as

$$\boldsymbol{Z}_1\boldsymbol{Z}_2, \quad \boldsymbol{Z}_2\boldsymbol{Z}_3 \quad \boldsymbol{Z}_4\boldsymbol{Z}_5 \quad \boldsymbol{Z}_5\boldsymbol{Z}_6, \quad \boldsymbol{Z}_7\boldsymbol{Z}_8 \quad \boldsymbol{Z}_8\boldsymbol{Z}_9$$

$$\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3\boldsymbol{X}_4\boldsymbol{X}_5\boldsymbol{X}_6, \quad \boldsymbol{X}_4\boldsymbol{X}_5\boldsymbol{X}_6\boldsymbol{X}_7\boldsymbol{X}_8\boldsymbol{X}_9. \tag{7.138}$$

In the notation of eq. (7.136) these become

$$\left(\begin{array}{ccc|ccc|ccc|ccccccccc}
1&1&0&\multicolumn{3}{c|}{0}&\multicolumn{3}{c|}{0}&\multicolumn{9}{c}{}\\
0&1&1&\multicolumn{3}{c|}{}&\multicolumn{3}{c|}{}&\multicolumn{9}{c}{}\\
\hline
\multicolumn{3}{c|}{0}&1&1&0&\multicolumn{3}{c|}{0}&\multicolumn{9}{c}{0}\\
\multicolumn{3}{c|}{}&0&1&1&\multicolumn{3}{c|}{}&\multicolumn{9}{c}{}\\
\hline
\multicolumn{3}{c|}{0}&\multicolumn{3}{c|}{0}&1&1&0&\multicolumn{9}{c}{}\\
\multicolumn{3}{c|}{}&\multicolumn{3}{c|}{}&0&1&1&\multicolumn{9}{c}{}\\
\hline
\multicolumn{6}{c|}{0}&\multicolumn{3}{c|}{}&1&1&1&1&1&1&0&0&0\\
\multicolumn{6}{c|}{}&\multicolumn{3}{c|}{}&0&0&0&1&1&1&1&1&1
\end{array}\right)$$

**(b) The seven-qubit code**. This $[[7,1,3]]$ code has six stabilizer generators, which can be expressed as

$$\tilde{H} = \left(\begin{array}{c|c} H_{\text{ham}} & 0 \\ 0 & H_{\text{ham}} \end{array}\right), \tag{7.139}$$

where $H_{\text{ham}}$ is the $3 \times 7$ parity-check matrix of the classical [7,4,3] Hamming code. The three check operators

$$\begin{aligned} \boldsymbol{M}_1 &= \boldsymbol{Z}_1\boldsymbol{Z}_3\boldsymbol{Z}_5\boldsymbol{Z}_7 \\ \boldsymbol{M}_2 &= \boldsymbol{Z}_2\boldsymbol{Z}_3\boldsymbol{Z}_6\boldsymbol{Z}_7 \\ \boldsymbol{M}_3 &= \boldsymbol{Z}_4\boldsymbol{Z}_5\boldsymbol{Z}_6\boldsymbol{Z}_7, \end{aligned} \tag{7.140}$$

detect the bit flips, and the three check operators

$$\begin{aligned} \boldsymbol{M}_4 &= \boldsymbol{X}_1\boldsymbol{X}_3\boldsymbol{X}_5\boldsymbol{X}_7 \\ \boldsymbol{M}_5 &= \boldsymbol{X}_2\boldsymbol{X}_3\boldsymbol{X}_6\boldsymbol{X}_7 \\ \boldsymbol{M}_6 &= \boldsymbol{X}_4\boldsymbol{X}_5\boldsymbol{X}_6\boldsymbol{X}_7, \end{aligned} \tag{7.141}$$

detect the phase errors. The space with $\boldsymbol{M}_1 = \boldsymbol{M}_2 = \boldsymbol{M}_3 = 1$ is spanned by the codewords that satisfy the Hamming parity check. Recalling that a Hadamard change of basis interchanges $\boldsymbol{Z}$ and $\boldsymbol{X}$, we see that the space with $\boldsymbol{M}_4 = \boldsymbol{M}_5 = \boldsymbol{M}_6$ is spanned by codewords that satisfy the Hamming parity check in the Hadamard-rotated basis. Indeed, we constructed the seven-qubit code by demanding that the Hamming parity check be satisfied in both bases. The generators commute because the Hamming code contains its dual code; *i.e.*, each row of $H_{\text{ham}}$ satisfies the Hamming parity check.

**(c) CSS codes**. Recall whenever an $[n,k,d]$ classical code $C$ contains its dual code $C^\perp$, we can perform the CSS construction to obtain an $[[n, 2k-n, d]]$ quantum code. The stabilizer of this code can be written as

$$\tilde{H} = \left(\begin{array}{c|c} H & 0 \\ 0 & H \end{array}\right) \tag{7.142}$$

where $H$ is the $(n-k) \times n$ parity check matrix of $C$. As for the seven-qubit code, the stabilizers commute because $C$ contains $C^\perp$, and the code subspace

is spanned by states that satisfy the $H$ parity check in both the $F$-basis and the $P$-basis. Equivalently, codewords obey the $H$ parity check and are invariant under

$$|v\rangle \to |v + w\rangle, \tag{7.143}$$

where $w \in C^{\perp}$.

**(d) More general CSS codes**. Consider, more generally, a stabilizer whose generators can each be chosen to be either a product of $\boldsymbol{Z}$'s $(\alpha|0)$ or a product of $\boldsymbol{X}$'s $(0|\beta)$. Then the generators have the form

$$\tilde{H} = \left( \begin{array}{c|c} H_Z & 0 \\ 0 & H_X \end{array} \right). \tag{7.144}$$

Now, what condition must $H_X$ and $H_Z$ satisfy if the $\boldsymbol{Z}$-generators and $\boldsymbol{X}$-generators are to commute? Since $\boldsymbol{Z}$'s must collide with $\boldsymbol{X}$'s an even number of times, we have

$$H_X H_Z^T = H_Z H_X^T = 0 \ . \tag{7.145}$$

But this is just the requirement that the dual $C_X^{\perp}$ of the code whose parity check is $H_X$ be contained in the code $C_Z$ whose parity check is $H_Z$. In other words, this QECC fits into the CSS framework, with

$$C_2 = C_X^{\perp} \subseteq C_1 = C_Z. \tag{7.146}$$

So we may characterize CSS codes as those and only those for which the stabilizer has generators of the form eq. (7.144).

However there is a caveat. The code defined by eq. (7.144) will be nondegenerate if errors are restricted to weight less than $d = \min(d_Z, d_X)$ (where $d_Z$ is the distance of $C_Z$, and $d_X$ the distance of $C_X$). But the true distance of the QECC could exceed $d$. For example, the 9-qubit code is in this generalized sense a CSS code. But in that case the classical code $C_X$ is distance 1, reflecting that, *e.g.*, $\boldsymbol{Z}_1\boldsymbol{Z}_2$ is contained in the stabilizer. Nevertheless, the distance of the CSS code is $d = 3$, since no weight-2 Pauli operator lies in $S^{\perp} \setminus S$.

### *7.10.4 Encoded qubits*

We have seen that the troublesome errors are those in $S^{\perp} \setminus S$ — those that commute with the stabilizer, but lie outside of it. These Pauli operators are also of interest for another reason: they can be regarded as the "logical" operations that act on the encoded data that is protected by the code.

Appealing to the "linear algebra" viewpoint, we can see that the normalizer $S^{\perp}$ of the stabilizer contains $n + k$ independent generators – in the $2n$-dimensional space of the $(\alpha|\beta)$'s, the subspace containing the vectors that are orthogonal to each of $n - k$ linearly independent vectors has dimension $2n - (n - k) = n + k$. Of the $n + k$ vectors that span this space, $n - k$ can be chosen to be the generators of the stabilizer itself. The remaining $2k$ generators preserve the code subspace because they commute with the stabilizer, but act nontrivially on the $k$ encoded qubits.

In fact, these $2k$ operations can be chosen to be the single-qubit operators $\bar{\boldsymbol{Z}}_i, \bar{\boldsymbol{X}}_i, i = 1, 2, \ldots, k$, where $\bar{\boldsymbol{Z}}_i, \bar{\boldsymbol{X}}_i$ are the Pauli operators $\boldsymbol{Z}$ and $\boldsymbol{X}$ acting on the encoded qubit labeled by $i$. First, note that we can extend the $n-k$ stabilizer generators to a maximal

set of $n$ commuting operators. The $k$ operators that we add to the set may be denoted $\bar{\boldsymbol{Z}}_1, \ldots \bar{\boldsymbol{Z}}_k$. We can then regard the simultaneous eigenstates of $\bar{\boldsymbol{Z}}_1 \ldots \bar{\boldsymbol{Z}}_k$ (in the code subspace $\mathcal{H}_S$) as the logical basis states $|\bar{z}_1, \ldots, \bar{z}_k\rangle$, with $\bar{z}_j = 0$ corresponding to $\bar{\boldsymbol{Z}}_j = 1$ and $\bar{z}_j = 1$ corresponding to $\bar{\boldsymbol{Z}}_j = -1$.

The remaining $k$ generators of the normalizer may be chosen to be mutually commuting and to commute with the stabilizer, but then they will not commute with any of the $\bar{\boldsymbol{Z}}_i$'s. By invoking a Gram-Schmidt orthonormalization procedure, we can choose these generators, denoted $\bar{\boldsymbol{X}}_i$, to diagonalize the symplectic form, so that

$$\bar{\boldsymbol{Z}}_i \bar{\boldsymbol{X}}_j = (-1)^{\delta_{ij}} \bar{\boldsymbol{X}}_j \bar{\boldsymbol{Z}}_i. \tag{7.147}$$

Thus, each $\bar{\boldsymbol{X}}_j$ flips the eigenvalue of the corresponding $\bar{\boldsymbol{Z}}_j$, and it can so be regarded as the Pauli operator $\boldsymbol{X}$ acting on encoded qubit $i$

**(a) The 9-qubit Code**. As we have discussed previously, the logical operators can be chosen to be

$$\begin{aligned} \bar{\boldsymbol{Z}} &= \boldsymbol{X}_1 \boldsymbol{X}_2 \boldsymbol{X}_3 \ , \\ \bar{\boldsymbol{X}} &= \boldsymbol{Z}_1 \boldsymbol{Z}_4 \boldsymbol{Z}_7 \ . \end{aligned} \tag{7.148}$$

These anti-commute with one another (an $\boldsymbol{X}$ and a $\boldsymbol{Z}$ collide at position 1), commute with the stabilizer generators, and are independent of the generators (no element of the stabilizer contains three $\boldsymbol{X}$'s or three $\boldsymbol{Z}$'s).

**(b) The 7-qubit code**. We have seen that

$$\begin{aligned} \bar{\boldsymbol{X}} &= \boldsymbol{X}_1 \boldsymbol{X}_2 \boldsymbol{X}_3 \ , \\ \bar{\boldsymbol{Z}} &= \boldsymbol{Z}_1 \boldsymbol{Z}_2 \boldsymbol{Z}_3 \ ; \end{aligned} \tag{7.149}$$

then $\bar{\boldsymbol{X}}$ adds an odd Hamming codeword and $\bar{\boldsymbol{Z}}$ flips the phase of an odd Hamming codeword. These operations implement a bit flip and phase flip respectively in the basis $\{|0\rangle_F, |1\rangle_F\}$ defined in eq. (7.102).

## 7.11 The 5-Qubit Code

All of the QECC's that we have considered so far are of the CSS type — each stabilizer generator is either a product of $\boldsymbol{Z}$'s or a product of $\boldsymbol{X}$'s. But not all stabilizer codes have this property. An example of a non-CSS stabilizer code is the perfect nondegenerate [[5,1,3]] code.

Its four stabilizer generators can be expressed

$$\begin{aligned} \boldsymbol{M}_1 &= \boldsymbol{XZZXI}, \\ \boldsymbol{M}_2 &= \boldsymbol{IXZZX}, \\ \boldsymbol{M}_3 &= \boldsymbol{XIXZZ}, \\ \boldsymbol{M}_4 &= \boldsymbol{ZXIXZ}, \end{aligned} \tag{7.150}$$

$\boldsymbol{M}_{2,3,4}$ are obtained from $\boldsymbol{M}_1$ by performing a cyclic permutation of the qubits. (The fifth operator obtained by a cyclic permutation of the qubits, $\boldsymbol{M}_5 = \boldsymbol{ZZXIX} = \boldsymbol{M}_1\boldsymbol{M}_2\boldsymbol{M}_3\boldsymbol{M}_4$ is not independent of the other four.) Since a cyclic permutation of a generator is another generator, the code itself is cyclic — a cyclic permutation of a codeword is a codeword.

Clearly each $\boldsymbol{M}_i$ contains no $\boldsymbol{Y}$'s and so squares to $\boldsymbol{I}$. For each pair of generators,

there are two collisions between an $\boldsymbol{X}$ and a $\boldsymbol{Z}$, so that the generators commute. One can quickly check that each Pauli operator of weight 1 or weight 2 anti-commutes with at least one generator, so that the distance of the code is 3.

Consider, for example, whether there are error operators with support on the first two qubits that commute with all four generators. The weight-2 operator, to commute with the $\boldsymbol{IX}$ in $\boldsymbol{M}_2$ and the $\boldsymbol{XI}$ in $\boldsymbol{M}_3$, must be $\boldsymbol{XX}$. But $\boldsymbol{XX}$ anti-commutes with the $\boldsymbol{XZ}$ in $\boldsymbol{M}_1$ and the $\boldsymbol{ZX}$ in $\boldsymbol{M}_4$.

In the symplectic notation, the stabilizer may be represented as

$$\tilde{H} = \left(\begin{array}{c|c} 01100 & 10010 \\ 00110 & 01001 \\ 00011 & 10100 \\ 10001 & 01010 \end{array}\right) \tag{7.151}$$

This matrix has a nice interpretation, as each of its columns can be regarded as the *syndrome* of a single-qubit error. For example, the single-qubit bit flip operator $\boldsymbol{X}_j$, commutes with $\boldsymbol{M}_i$ if $\boldsymbol{M}_i$ has an $\boldsymbol{I}$ or $\boldsymbol{X}$ in position $j$, and anti-commutes if $\boldsymbol{M}_i$ has a $\boldsymbol{Z}$ in position $j$. Thus the table

| | $\boldsymbol{X}_1$ | $\boldsymbol{X}_2$ | $\boldsymbol{X}_3$ | $\boldsymbol{X}_4$ | $\boldsymbol{X}_5$ |
|---|---|---|---|---|---|
| $\boldsymbol{M}_1$ | 0 | 1 | 1 | 0 | 0 |
| $\boldsymbol{M}_2$ | 0 | 0 | 1 | 1 | 0 |
| $\boldsymbol{M}_3$ | 0 | 0 | 0 | 1 | 1 |
| $\boldsymbol{M}_4$ | 1 | 0 | 0 | 0 | 1 |

lists the outcome of measuring $\boldsymbol{M}_{1,2,3,4}$ in the event of a bit flip. (For example, if the first bit flips, the measurement outcomes $\boldsymbol{M}_1 = \boldsymbol{M}_2 = \boldsymbol{M}_3 = 1, \boldsymbol{M}_4 = -1$, diagnose the error.) Similarly, the right half of $\tilde{H}$ can be regarded as the syndrome table for the phase errors.

| | $\boldsymbol{Z}_1$ | $\boldsymbol{Z}_2$ | $\boldsymbol{Z}_3$ | $\boldsymbol{Z}_4$ | $\boldsymbol{Z}_5$ |
|---|---|---|---|---|---|
| $\boldsymbol{M}_1$ | 1 | 0 | 0 | 1 | 0 |
| $\boldsymbol{M}_2$ | 0 | 1 | 0 | 0 | 1 |
| $\boldsymbol{M}_3$ | 1 | 0 | 1 | 0 | 0 |
| $\boldsymbol{M}_4$ | 0 | 1 | 0 | 1 | 0 |

Since $\boldsymbol{Y}$ anti-commutes with both $\boldsymbol{X}$ and $\boldsymbol{Z}$, we obtain the syndrome for the error $\boldsymbol{Y}_i$ by summing the $i$th columns of the $\boldsymbol{X}$ and $\boldsymbol{Z}$ tables:

| | $\boldsymbol{Y}_1$ | $\boldsymbol{Y}_2$ | $\boldsymbol{Y}_3$ | $\boldsymbol{Y}_4$ | $\boldsymbol{Y}_5$ |
|---|---|---|---|---|---|
| $\boldsymbol{M}_1$ | 1 | 1 | 1 | 1 | 0 |
| $\boldsymbol{M}_2$ | 0 | 1 | 1 | 1 | 1 |
| $\boldsymbol{M}_3$ | 1 | 0 | 1 | 1 | 1 |
| $\boldsymbol{M}_4$ | 1 | 1 | 0 | 1 | 1 |

We find by inspection that the 15 columns of the $\boldsymbol{X}, \boldsymbol{Y}$, and $\boldsymbol{Z}$ syndrome tables are all distinct, and so we verify again that our code is a nondegenerate code that corrects one error. Indeed, the code is perfect — each of the 15 nontrivial binary strings of length 4 appears as a column in one of the tables.

Because of the cyclic property of the code, we can easily characterize all 15 nontrivial

elements of its stabilizer. Aside from $\boldsymbol{M}_1 = \boldsymbol{XZZXI}$ and the four operators obtained from it by cyclic permutations of the qubit, the stabilizer also contains

$$\boldsymbol{M}_3\boldsymbol{M}_4 = -\boldsymbol{YXXYI}, \tag{7.152}$$

plus its cyclic permutations, and

$$\boldsymbol{M}_2\boldsymbol{M}_5 = -\boldsymbol{ZYYZI}, \tag{7.153}$$

and its cyclic permutations. Evidently, all elements of the stabilizer are weight-4 Pauli operators.

For our logical operators, we may choose

$$\begin{aligned} \bar{\boldsymbol{Z}} &= \boldsymbol{ZZZZZ}, \\ \bar{\boldsymbol{X}} &= \boldsymbol{XXXXX}; \end{aligned} \tag{7.154}$$

these commute with $\boldsymbol{M}_{1,2,3,4}$, square to $\boldsymbol{I}$, and anti-commute with one another. Being weight 5, they are not themselves contained in the stabilizer. Therefore if we don't mind destroying the encoded state, we can determine the value of $\bar{\boldsymbol{Z}}$ for the encoded qubit by measuring $\boldsymbol{Z}$ of each qubit and evaluating the parity of the outcomes. In fact, since the code is distance three, there are elements of $S^\perp \setminus S$ of weight-three; alternate expressions for $\bar{\boldsymbol{Z}}$ and $\bar{\boldsymbol{X}}$ can be obtained by multiplying by elements of the stabilizer. For example we can choose

$$\bar{\boldsymbol{Z}} = (\boldsymbol{ZZZZZ}) \cdot (-\boldsymbol{ZYYZI}) = -\boldsymbol{IXXIZ}, \tag{7.155}$$

(or one of its cyclic permutations), and

$$\bar{\boldsymbol{X}} = (\boldsymbol{XXXXX}) \cdot (-\boldsymbol{YXXYI}) = -\boldsymbol{ZIIZX}, \tag{7.156}$$

(or one of its cyclic permutations). So it is possible to ascertain the value of $\bar{\boldsymbol{X}}$ or $\bar{\boldsymbol{Z}}$ by measuring $\boldsymbol{X}$ or $\boldsymbol{Z}$ of only three of the five qubits in the block, and evaluating the parity of the outcomes.

If we wish, we can construct an orthonormal basis for the code subspace, as follows. Starting from any state $|\psi_0\rangle$, we can obtain

$$|\Psi_0\rangle = \sum_{\boldsymbol{M}\in S} \boldsymbol{M}|\psi_0\rangle. \tag{7.157}$$

This (unnormalized) state obeys $\boldsymbol{M}'|\Psi_0\rangle = |\Psi_0\rangle$ for each $\boldsymbol{M}' \in S$, since multiplication by an element of the stabilizer merely permutes the terms in the sum. To obtain the $\bar{\boldsymbol{Z}} = 1$ encoded state $|\bar{0}\rangle$, we may start with the state $|00000\rangle$, which is also a $\bar{\boldsymbol{Z}} = 1$ eigenstate, but not in the stabilizer; we find (up to normalization)

$$\begin{aligned} |\bar{0}\rangle &= \sum_{\boldsymbol{M}\in S} |00000\rangle \\ &= |00000\rangle + (\boldsymbol{M}_1 + \text{cyclic perms})\,|00000\rangle \\ &+ (\boldsymbol{M}_3\boldsymbol{M}_4 + \text{cyclic perms})\,|00000\rangle + (\boldsymbol{M}_2\boldsymbol{M}_5 + \text{cyclic perms})\,|00000\rangle \\ &= |00000\rangle + (110010\rangle + \text{ cyclic perms}) \\ &- (|11110\rangle + \text{ cyclic perms}) \\ &- (|01100\rangle + \text{ cyclic perms}). \end{aligned} \tag{7.158}$$

We may then find $|\bar{1}\rangle$ by applying $\bar{\boldsymbol{X}}$ to $|\bar{0}\rangle$, that is by flipping all 5 qubits:

$$\begin{aligned}|\bar{1}\rangle = \bar{\boldsymbol{X}}|\bar{0}\rangle = |11111\rangle &+ (|01101\rangle + \text{ cyclic perms}) \\ &- (|00001\rangle + \text{ cyclic perms}) \\ &- (|10011\rangle + \text{ cyclic perms}) \ . \end{aligned} \tag{7.159}$$

How is the syndrome measured? A circuit that can be executed to measure $\boldsymbol{M}_1 = \boldsymbol{XZZXI}$ is:

– Figure –

The Hadamard rotations on the first and fourth qubits rotate $\boldsymbol{M}_1$ to the tensor product of $\boldsymbol{Z}$'s $\boldsymbol{ZZZZI}$, and the CNOT's then imprint the value of this operator on the ancilla. The final Hadamard rotations return the encoded block to the standard code subspace. Circuits for measuring $\boldsymbol{M}_{2,3,4}$ are obtained from the above by cyclically permuting the five qubits in the code block.

What about encoding? We want to construct a unitary transformation

$$\boldsymbol{U}_{\text{encode}} : |0000\rangle \otimes (a|0\rangle + b|1\rangle) \to a|\bar{0}\rangle + b|\bar{1}\rangle. \tag{7.160}$$

We have already seen that $|00000\rangle$ is a $\bar{\boldsymbol{Z}} = 1$ eigenstate, and that $|00001\rangle$ is a $\bar{\boldsymbol{Z}} = -1$ eigenstate. Therefore (up to normalization)

$$a|\bar{0}\rangle + b|\bar{1}\rangle = \left(\sum_{\boldsymbol{M}\in S} \boldsymbol{M}\right) |0000\rangle \otimes (a|0\rangle + b|1\rangle). \tag{7.161}$$

So we need to figure out how to construct a circuit that applies $(\sum \boldsymbol{M})$ to an initial state.

Since the generators are independent, each element of the stabilizer can be expressed as a product of generators as a unique way, and we may therefore rewrite the sum as

$$\sum_{\boldsymbol{M}\in S} \boldsymbol{M} = (\boldsymbol{I} + \boldsymbol{M}_4)(\boldsymbol{I} + \boldsymbol{M}_3)(\boldsymbol{I} + \boldsymbol{M}_2)(\boldsymbol{I} + \boldsymbol{M}_1) \ . \tag{7.162}$$

Now to proceed further it is convenient to express the stabilizer in an alternative form. Note that we have the freedom to replace the generator $\boldsymbol{M}_i$ by $\boldsymbol{M}_i\boldsymbol{M}_j$ without changing the stabilizer. This replacement is equivalent to adding the $j$th row to the $i$th row in the matrix $\tilde{H}$. With such row operations, we can perform a Gaussian elimination on the $4 \times 5$ matrix $H_X$, and so obtain the new presentation for the stabilizer

$$\tilde{H}' = \left(\begin{array}{c|c} 11011 & 10001 \\ 00110 & 01001 \\ 11000 & 00101 \\ 10111 & 00011 \end{array}\right) \ , \tag{7.163}$$

or

$$\begin{aligned} \boldsymbol{M}_1 &= \boldsymbol{YZIZY} \\ \boldsymbol{M}_2 &= \boldsymbol{IXZZX} \\ \boldsymbol{M}_3 &= \boldsymbol{ZZXIX} \\ \boldsymbol{M}_4 &= \boldsymbol{ZIZYY} \end{aligned} \tag{7.164}$$

In this form $\boldsymbol{M}_i$ applies an $\boldsymbol{X}$ (flip) only to qubits $i$ and 5 in the block.

Adopting this form for the stabilizer, we can apply $\frac{1}{\sqrt{2}}(\boldsymbol{I} + \boldsymbol{M}_1)$ to a state $|0, z_2, z_3, z_4, z_5\rangle$ by executing the circuit

– Figure –

The Hadamard prepares $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle$. If the first qubit is $|0\rangle$, the other operations don't do anything, so $\boldsymbol{I}$ is applied. But if the first qubit is $|1\rangle$, then $\boldsymbol{X}$ has been applied to this qubit, and the other gates in the circuit apply $\boldsymbol{ZZIZY}$, conditioned on the first qubit being $|1\rangle$. Hence, $\boldsymbol{YZIZY} = \boldsymbol{M}_1$ has been applied. Similar circuits can be constructed that apply $\frac{1}{\sqrt{2}}(\boldsymbol{I}+\boldsymbol{M}_2)$ to $|z_1, 0, z_3, z_4, z_5\rangle$, and so forth. Apart from the Hadamard gates each of these circuits applies only $\boldsymbol{Z}$'s and conditional $\boldsymbol{Z}$'s to qubits 1 through 4; these qubits never flip. (It was to ensure thus that we performed the Gaussian elimination on $H_X$.) Therefore, we can construct our encoding circuit as

– Figure –

Furthermore, each $\boldsymbol{Z}$ gate acting on $|0\rangle$ can be replaced by the identity, so we may simplify the circuit by eliminating all such gates, obtaining

– Figure –

This procedure can be generalized to construct an encoding circuit for any stabilizer code.

Since the encoding transformation is unitary, we can use its adjoint to decode. And since each gate squares to $\pm\boldsymbol{I}$, the decoding circuit is just the encoding circuit run in reverse.

## 7.12 Quantum secret sharing

The $[[5, 1, 3]]$ code provides a nice illustration of a possible application of QECC's.[2]

Suppose that some top secret information is to be entrusted to $n$ parties. Because none is entirely trusted, the secret is divided into $n$ shares, so that each party, with access to his share alone, can learn nothing at all about the secret. But if enough parties get together and pool their shares, they can decipher the secret or some part of it.

In particular, an $(m, n)$ threshold scheme has the property that $m$ shares are sufficient to reconstruct all of the secret information. But from $m-1$ shares, no information at all can be extracted. (This is called a *threshold* scheme because as shares $1, 2, 3 \ldots, m-1$ are collected one by one, nothing is learned, but the next share crosses the threshold and reveals everything.)

[2] R. Cleve, D. Gottesman, and H.-K. Lo, "How to Share a Quantum Secret," quant-ph/9901025.

We should distinguish too kinds of secrets: a classical secret is an *a priori* unknown bit string, while a quantum secret is an *a priori* unknown quantum state. Either type of secret can be shared. In particular, we can distribute a classical secret among several parties by selecting one from an ensemble of mutually orthogonal (entangled) quantum states, and dividing the state among the parties.

We can see, for example, that the $[[5, 1, 3]]$ code may be employed in a $(3, 5)$ threshold scheme, where the shared information is classical. One classical bit is encoded by preparing one of the two orthogonal states $|\bar{0}\rangle$ or $|\bar{1}\rangle$ and then the five qubits are distributed to five parties. We have seen that (since the code is nondegenerate) if any two parties get together, then the density matrix $\boldsymbol{\rho}$ their two qubits is

$$\boldsymbol{\rho}^{(2)} = \frac{1}{4}\mathbf{1} \ . \tag{7.165}$$

Hence, they learn nothing about the quantum state from any measurement of their two qubits. But we have also seen that the code can correct two located errors or two erasures. When any three parties get together, they may correct the two errors (the two missing qubits) and perfectly reconstruct the encoded state $|\bar{0}\rangle$ or $|\bar{1}\rangle$.

It is also clear that by a similar procedure a single qubit of quantum information can be shared – the $[[5, 1, 3]]$ code is also the basis of a $((3, 5))$ quantum threshold scheme (we use the $((m, n))$ notation if the shared information is quantum information, and the $(m, n)$ notation if the shared information is classical). How does this quantum-secret-sharing scenario generalize to more qubits? Suppose we prepare a pure state $|\psi\rangle$ of $n$ qubits — can it be employed in an $((m, n))$ threshold scheme?

We know that $m$ qubits must be sufficient to reconstruct the state; hence $n - m$ erasures can be corrected. It follows from our general error correction criterion that the expectation value of any weight-$(n - m)$ observable must be independent of the state $|\psi\rangle$

$$\langle\psi|\boldsymbol{E}|\psi\rangle \text{ independent of } |\psi\rangle, \quad \mathrm{wt}(\boldsymbol{E}) \leq n - m. \tag{7.166}$$

Thus, if $m$ parties have all the information, the other $n-m$ parties have *no* information at all. That makes sense, since quantum information cannot be cloned.

On the other hand, we know that $m - 1$ shares reveal nothing, or that

$$\langle\psi|\boldsymbol{E}|\psi\rangle \text{ independent of } |\psi\rangle, \quad \mathrm{wt}(\boldsymbol{E}) \leq m - 1. \tag{7.167}$$

It then follows that $m-1$ erasures can be corrected, or that the other $n-m+1$ parties have all the information.

From these two observations we obtain the two inequalities

$$\begin{aligned} n - m < m \Rightarrow n < 2m \ , \\ m - 1 < n - m + 1 \Rightarrow n > 2m - 2 \ . \end{aligned} \tag{7.168}$$

It follows that

$$n = 2m - 1 \ , \tag{7.169}$$

in an $((m, n))$ pure state quantum threshold scheme, where each party has a single qubit. In other words, the threshold is reached as the number of qubits in hand crosses over from the minority to the majority of all $n$ qubits.

We see that if each share is a qubit, a quantum pure state threshold scheme is a $[[2m - 1, k, m]]$ quantum code with $k \geq 1$. But in fact the $[[3, 1, 2]]$ and $[[7, 1, 4]]$ codes

do not exist, and it follows from the Rains bound that the $m > 3$ codes do not exist. In a sense, then, the $[[5, 1, 3]]$ code is the unique quantum threshold scheme.

There are a number of caveats — the restriction $n = 2m - 1$ continues to apply if each share is a $q$-dimensional system rather than a qubit, but various

$$[[2m - 1, 1, k]]_q \tag{7.170}$$

codes can be constructed for $q > 2$. (See the exercises for an example.)

Also, we might allow the shared information to be a mixed state (that encodes a pure state). For example, if we discard one qubit of the five qubit block, we have a $((3, 4))$ scheme. Again, once we have three qubits, we can correct two erasures, one arising because the fourth share is in the hands of another party, the other arising because a qubit has been thrown away.

Finally, we have assumed that the shared information is quantum information. But if we are only sharing classical information instead, then the conditions for correcting erasures are less stringent. For example, a Bell pair may be regarded as a kind of $(2, 2)$ threshold scheme for two bits of classical information, where the classical information is encoded by choosing one of the four mutually orthogonal states $|\phi^{\pm}\rangle, |\psi^{\pm}\rangle$. A party in possession of one of the two qubits is unable to access any of this classical information. But this is not a scheme for sharing a quantum secret, since linear combinations of these Bell states do *not* have the property that $\rho = \frac{1}{2}\mathbf{1}$ if we trace out one of the two qubits.

## 7.13 Some Other Stabilizer Codes

### *7.13.1 The* $[[6, 0, 4]]$ *code*

A $k = 0$ quantum code has a one-dimensional code subspace; that is, there is only one encoded state. The code cannot be used to store unknown quantum information, but even so, $k = 0$ codes can have interesting properties. Since they can detect and diagnose errors, they might be useful for a study of the correlations in decoherence induced by interactions with the environment.

If $k = 0$, then $S$ and $S^{\perp}$ coincide – a Pauli operator that commutes with all elements of the stabilizer must lie in the stabilizer. In this case, the distance $d$ is defined as the minimum weight of any Pauli operator in the stabilizer. Thus a distance-$d$ code can "detect $d-1$ errors;" that is, if any Pauli operator of weight less than $d$ acts on the code state, the result is orthogonal to that state.

Associated with the $[[5, 1, 3]]$ code is a $[[6, 0, 4]]$ code, whose encoded state can be expressed as

$$|0\rangle \otimes |\bar{0}\rangle + |1\rangle \otimes |\bar{1}\rangle, \tag{7.171}$$

where $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are the $\bar{\boldsymbol{Z}}$ eigenstates of the $[[5, 1, 3]]$ code. You can verify that this code has distance $d = 4$ (an exercise).

The $[[6, 0, 4]]$ code is interesting because its code state is maximally entangled. We may choose any three qubits from among the six. The density matrix $\boldsymbol{\rho}^{(3)}$ of those three, obtained by tracing over the other three, is totally random, $\boldsymbol{\rho}^{(3)} = \frac{1}{8}\boldsymbol{I}$. In this sense, the $[[6, 0, 4]]$ state is a natural multiparticle analog of the two-qubit Bell states. It is far "more entangled" than the six-qubit cat state $\frac{1}{\sqrt{2}}(|000000\rangle + |111111\rangle)$. If we measure any one of the six qubits in the cat state, in the $\{|0\rangle, |1\rangle\}$ basis, we know everything about the state we have prepared of the remaining five qubits. But we may measure

any observable we please acting on any *three* qubits in the $[[6,0,4]]$ state, and we learn *nothing* about the remaining three qubits, which are still described by $\boldsymbol{\rho}^{(3)} = \frac{1}{8}\boldsymbol{I}$.

Our $[[6,0,4]]$ state is all the more interesting in that it turns out (but is not so simple to prove) that its generalizations to more qubits do not exist. That is, there are no $[[2n,0,n+1]]$ binary quantum codes for $n > 3$. You'll see in the exercises, though, that there are other, nonbinary, maximally entangled states that can be constructed.

### 7.13.2 The $[[2m, 2m-2, 2]]$ error-detecting codes

The Bell state $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is a $[[2,0,2]]$ code with stabilizer generators

$$\begin{array}{c} \boldsymbol{ZZ}\ , \\ \boldsymbol{XX}\ . \end{array} \tag{7.172}$$

The code has distance two because no weight-one Pauli operator commutes with both generators (none of $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ commute with both $\boldsymbol{X}$ and $\boldsymbol{Z}$). Correspondingly, a bit flip ($\boldsymbol{X}$) or a phase flip ($\boldsymbol{Z}$), or both ($\boldsymbol{Y}$) acting on either qubit in $|\phi^+\rangle$, takes it to an orthogonal state (one of the other Bell states $|\phi^-\rangle, |\psi^+\rangle, |\psi^-\rangle$).

One way to generalize the Bell states to more qubits is to consider the $n = 4, k = 2$ code with stabilizer generators

$$\begin{array}{c} \boldsymbol{ZZZZ}\ , \\ \boldsymbol{XXXX}\ . \end{array} \tag{7.173}$$

This is a distance $d = 2$ code for the same reason as before. The code subspace is spanned by states of even parity ($\boldsymbol{ZZZZ}$) that are invariant under a simultaneous flip of all four qubits ($\boldsymbol{XXX}$). A basis is:

$$\begin{array}{ccc} |0000\rangle & + & |1111\rangle\ , \\ |0011\rangle & + & |1100\rangle\ , \\ |0101\rangle & + & |1010\rangle\ , \\ |0110\rangle & + & |1001\rangle\ . \end{array} \tag{7.174}$$

Evidently, an $\boldsymbol{X}$ or a $\boldsymbol{Z}$ acting on any qubit takes each of these states to a state orthogonal to the code subspace; thus any single-qubit error can be detected.

A further generalization is the $[[2m, 2m-2, 2]]$ code with stabilizer generators

$$\begin{array}{ccc} \boldsymbol{ZZ} & \ldots & \boldsymbol{Z}\ , \\ \boldsymbol{XX} & \ldots & \boldsymbol{X}\ , \end{array} \tag{7.175}$$

(the length is required to be even so that the generators will commute. The code subspace is spanned by our familiar friends the $2^{n-2}$ cat states

$$\frac{1}{\sqrt{2}}(|x\rangle + |\neg x\rangle), \tag{7.176}$$

where $x$ is an even-weight string of length $n = 2m$.

### 7.13.3 The $[[8,3,3]]$ code

As already noted in our discussion of the $[[5,1,3]]$ code, a stabilizer code with generators

$$\tilde{H} = (H_Z | H_X), \tag{7.177}$$

can correct one error if: (1) the columns of $\tilde{H}$ are distinct (a distinct syndrome for each $\boldsymbol{X}$ and $\boldsymbol{Z}$ error) and (2) each sum of a column of $H_Z$ with the corresponding column of $H_X$ is distinct from each column of $\tilde{H}$ and distinct from all other such sums (each $\boldsymbol{Y}$ error can be distinguished from all other one-qubit errors).

We can readily construct a $5 \times 16$ matrix $\tilde{H}$ with this property, and so derive the stabilizer of an $[[8, 3, 3]]$ code; we choose

$$\tilde{H} = \left( \begin{array}{c|c} H & H^\sigma \\ 11111111 & 00000000 \\ 00000000 & 11111111 \end{array} \right) . \tag{7.178}$$

Here $H$ is the $3 \times 8$ matrix

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{7.179}$$

whose columns are all the distinct binary strings of length 3, and $H^\sigma$ is obtained from $H$ by performing a suitable permutation of the columns. This permutation is chosen so that the eight sums of columns of $H$ with corresponding columns of $H^\sigma$ are all distinct. We may see by inspection that a suitable choice is

$$H^\sigma = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \tag{7.180}$$

as the column sums are then

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} . \tag{7.181}$$

The last two rows of $\tilde{H}$ serve to distinguish each $\boldsymbol{X}$ syndrome from each $\boldsymbol{Y}$ syndrome or $\boldsymbol{Z}$ syndrome, and the above mentioned property of $H^\sigma$ ensures that all $\boldsymbol{Y}$ syndromes are distinct. Therefore, we have constructed a length-8 code with $k = 8 - 5 = 3$ that can correct one error. It is actually the simplest in an infinite class of $[[2^m, 2^m - m - 2, 3]]$ codes constructed by Gottesman, with $m \geq 3$.

The $[[8, 3, 3]]$ quantum code that we have just described is a close cousin of the "extended Hamming code," the self-dual [8,4,4] classical code that is obtained from the [7,3,4] dual of the Hamming code by adding an extra parity bit. Its parity check matrix (which is also its generator matrix) is

$$H_{\mathrm{EH}} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{7.182}$$

This matrix $H_{\mathrm{EH}}$ has the property that, not only are its eight columns distinct, but also each *sum* of two columns is distinct from all columns; since the sum of two columns has 0, not 1, as its fourth bit.

## 7.14 Cleaning lemma for stabilizer codes

The *cleaning lemma* for stabilizer codes says the following: For an $[[n, k]]$ stabilizer code, let $M$ denote a subset of the $n$ qubits in the code block, and let $M^c$ denote the complementary set of qubits. If $x$ is one of the code's logical Pauli operators, we say that $x$ can be *cleaned* on $M$ if there is a logically equivalent Pauli operator $x' = xy$ (where $y$ is an element of the code stabilizer $S$) such that $x'$ acts nontrivially only on $M^c$:

$$x' = I_M \otimes Q_{M^c}. \tag{7.183}$$

We say that $x$ can be *supported* on $M$ if it can be cleaned on $M^c$. Let $g(M)$ denote the number of independent logical Pauli operators that can be supported on $M$ and let $g(M^c)$ denote the number of independent Pauli operators that can be supported on $M^c$. Then the cleaning lemma asserts that

$$g(M) + g(M^c) = 2k. \tag{7.184}$$

In particular, therefore, if no logical operator can be supported on $M$, then the complete $k$-qubit logical Pauli group can be supported on its complement.

We say that the subset $M$ is *correctable* if erasure of the qubits in $M$ is a correctable error. From the error correction conditions for stabilizer codes, we may say that if $M$ is correctable, then any Pauli operator supported on $M$ either anticommutes with the stabilizer or is contained in the stabilizer. Conversely, if $M$ is not correctable, then there is a nontrivial Pauli operator supported on $M$ which commutes with the stabilizer and is not contained in the statilizer; that is, if $M$ is not correctable, then there is a nontrivial logical operator supported on $M$.

To prove the cleaning lemma we proceed as follows. We regard the abelianized $n$-qubit Pauli group $P$ as a $(2n)$-dimensional vector space over the binary field $\mathbb{F}_2$, and say that the vectors $x$ and $y$ are orthogonal if the corresponding elements of $P$ commute. Let $P_M$ denote the subspace of $P$ which is supported on the subset $M$ of the $n$ qubits. Let $S$ denote the stabilizer of an $[[n, k]]$ quantum stabilizer code. Let $[T]$ denote the dimension of a subspace $T$.

We may express $S$ as

$$S = S_M \oplus S_{M^c} \oplus S'. \tag{7.185}$$

Here $S_M = S \cap P_M$ is the subspace of $S$ which is supported on $M$, $S_{M^c} = S \cap P_{M^c}$ is the subspace of $S$ which is supported on $M^c$, and $S'$ is a third subspace of $S$. For any vector $x$, we may consider its *restriction* $x|_M$ to the set $M$. For a Pauli operator $x = x_1 \otimes x_2$, with $x_1$ supported on $M$ and $x_2$ supported on $M^c$, its restriction to $M$ is $x_1 \otimes I$.

Now consider the restriction $S'|_M$ of $S'$ to $M$. We claim that $[S_M \oplus S'|_M] = [S_M \oplus S'] = [S_M] + [S']$. That is, linearly independent vectors in $S_M + S'$ remain linearly independent when restricted to $M$. If this were not so, then a nontrivial linear combination of these vectors would have a trivial restriction to $M$, and would therefore be contained in $S_{M^c}$.

Let's compute the dimension $[\left(S^\perp\right)_M]$ of $\left(S^\perp\right)_M = S^\perp \cap P_M$, where $S^\perp$ is the orthogonal complement of $S$ in $P$. Note that, because $S_{M^c}$ is trivially orthogonal to $P_M$, $\left(S^\perp\right)_M$ is the orthogonal complement in $P_M$ of the restriction $\left(S_M \oplus S'\right)|_M$ of $S_M \oplus S'$ to $M$. Therefore, by counting dimensions,

$$[\left(S^\perp\right)_M] = [P_M] - [S_M \oplus S'|_M] = [P_M] - [S_M] - [S']. \tag{7.186}$$

By applying the same reasoning with $M$ replaced by $M^c$, we have

$$[\left(S^\perp\right)_{M^c}] = [P_{M^c}] - [S_{M^c} \oplus S'|_{M^c}] = [P_{M^c}] - [S_{M^c}] - [S']. \tag{7.187}$$

Logical operators supported on $M$ are elements of $P_M$ which commute with the stabilizer and are not contained in the stabilizer (and same for $M^c$); therefore

$$\begin{aligned} g(M) &= [\left(S^\perp\right)_M] - [S_M] = [P_M] - 2[S_M] - [S'], \\ g(M^c) &= [\left(S^\perp\right)_{M^c}] - [S_{M^c}] = [P_{M^c}] - 2[S_{M^c}] - [S'], \end{aligned} \tag{7.188}$$

and hence

$$\begin{aligned} g(M) + g(M^c) &= [P_M] + [P_{M^c}] - 2\left([S_M] + [S_{M^c}] + [S']\right), \\ &= [P] - 2[S] = 2n - 2(n-k) = 2k, \end{aligned} \tag{7.189}$$

as we wanted to show.

An important caveat: For a logical operator $x \in S^\perp \setminus S$, there might exist $y$, $y'$ in $S$ such that $xy$ is supported on $M$ and $xy'$ is supported on $M^c$. Therefore, we may not conclude from eq.(7.329) that *each* of the code's $2k$ independent logical Pauli operators can be supported on either $M$ or $M^c$. Consider, for example, the toric code with $k = 2$, where $M$ is a narrow strip of qubits that winds around one cycle of the torus. This strip supports the code's string logical operators $X_1$ and $Z_2$, where 1,2 denote the two protected qubits, and $M^c$ also supports these same two logical operators; hence $g(M) = 2$ and $g(M^c) = 2$. But neither $M$ nor $M^c$ supports $X_2$ or $Z_1$, the string logical operators that wind around the complementary cycle of the torus.

If erasure of $M$ is correctable, then $g(M) = 0$ and we conclude that $g(M^c) = 2k$, so that all logical Pauli operators may be supported on $M^c$. We will use this consequence of the cleaning lemma in the arguments below.

## 7.15 Codes Over $GF(4)$

We constructed the $[[5, 1, 3]]$ code by guessing the stabilizer generators, and checking that $d = 3$. Is there a more systematic method?

In fact, there is. Our suspicion that the $[[5, 1, 3]]$ code might exist was aroused by the observation that its parameters saturate the quantum sphere-packing inequality for $t = 1$ codes:

$$1 + 3n = 2^{n-k}, \tag{7.190}$$

($16 = 16$ for $n = 5$ and $k = 1$). To a coding theorist, this equation might look familiar.

Aside from the binary codes we have focused on up to now, classical codes can also be constructed from length-$n$ strings of symbols that take values, not in $\{0, 1\}$, but in the finite field with $q$ elements $GF(q)$. Such finite fields exist for any $q = p^m$, where $p$ is prime. ($GF$ is short for "Galois Field," in honor of their discoverer.)

For such nonbinary codes, we may model error as addition by an element of the field, a cyclic shift of the $q$ symbols. Then there are $q - 1$ nontrivial errors. The weight of a vector in $GF(q)^n$ is the number of its nonzero elements, and the distance between two vectors is the weight of their difference (the number of elements that disagree). An $[n, k, d]_q$ classical code consists of $q^k$ codewords in $GF(q)^n$, where the minimal distance

between a pair is $d$. The sphere packing bound that must be satisfied for an $[n,k,d]_q$ code to exist becomes, for $d=3$,

$$1+(q-1)n \le q^{n-k}. \tag{7.191}$$

In fact, the perfect binary Hamming codes that saturate this bound for $q=2$ with parameters

$$n=2^m-1, \quad k=n-m, \tag{7.192}$$

admit a generalization to any $GF(q)$; perfect Hamming codes over $GF(q)$ can be constructed with

$$n=\frac{q^m-1}{q-1}\ , \quad k=n-m\ . \tag{7.193}$$

The $[[5,1,3]]$ quantum code is descended from the classical $[5,3,3]_4$ Hamming code (the case $q=4$ and $m=2$).

What do the classical $GF(4)$ codes have to do with binary quantum stabilizer codes? The connection arises because the stabilizer can be associated with a set of vectors over $GF(4)$ closed under addition.

The field $GF(4)$ has four elements that may be denoted $0,1,\omega,\bar{\omega}$, where

$$\begin{aligned} 1+1=\omega+\omega=\bar{\omega}+\bar{\omega}=0,\\ 1+\omega=\bar{\omega}, \end{aligned} \tag{7.194}$$

and $\omega^2=\bar{\omega}$, $\omega\bar{\omega}=1$. Thus, the additive structure of $GF(4)$ echos the multiplicative structure of the Pauli operators $\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}$. Indeed, the length-$2n$ binary string $(\alpha|\beta)$ that we have used to denote an element of the Pauli group can equivalently be regarded as a length-$n$ vector in $GF(4)^n$

$$(\alpha|\beta) \leftrightarrow \alpha+\beta\omega. \tag{7.195}$$

The stabilizer, with $2^{n-k}$ elements, can be regarded as a subcode of $GF(4)$, closed under addition and containing $2^{n-k}$ codewords.

Note that the code need not be a vector space over $GF(4)$, as it is not required to be closed under multiplication by a scalar $\in GF(4)$. In the special case where the code is a vector space, it is called a *linear* code.

Much is known about codes over $GF(4)$, so this connection opened the door for the (classical) coding theorists to construct many QECC's.[3] However, not every subcode of $GF(4)^n$ is associated with a quantum code; we have not yet imposed the requirement that the stabilizer is abelian – the $(\alpha|\beta)$'s that span the code must be mutually orthogonal in the symplectic inner product

$$\alpha\cdot\beta'+\alpha'\cdot\beta\ . \tag{7.196}$$

This orthogonality condition might look strange to a coding theorist, who is more accustomed to defining the inner product of two vectors in $GF(4)^n$ as an element of $GF(4)$ given by

$$v*u=\bar{v}_1u_1+\cdots+\bar{v}_nu_n\ , \tag{7.197}$$

where conjugation, denoted by a bar, interchanges $\omega$ and $\bar{\omega}$. If this "hermitian" inner product $*$ of two vectors $v$ and $u$ is

$$v*u=a+b\omega\in GF(4)\ , \tag{7.198}$$

[3] Calderbank, Rains, Shor, and Sloane, "Quantum error correction via codes over $GF(4)$," quant-ph/9608006.

then our symplectic inner product is

$$v \cdot u = b \ . \tag{7.199}$$

Therefore, vanishing of the symplectic inner product is a weaker condition than vanishing of the hermitian inner product. In fact, though, in the special case of a *linear* code, self-orthogonality with respect to the hermitian inner product is actually equivalent to self-orthogonality with respect to the symplectic inner product. We observe that if $v * u = a + b\omega$, orthogonality in the symplectic inner product requires $b = 0$. But if $u$ is in a linear code, then so is $\bar{\omega}u$ where

$$v * (\bar{\omega}u) = b + a\bar{\omega} \tag{7.200}$$

so that

$$v \cdot (\bar{\omega}u) = a \ . \tag{7.201}$$

We see that if $v$ and $u$ belong to a linear $GF(4)$ code and are orthogonal with respect to the symplectic inner product, then they are also orthogonal with respect to the hermitian inner product. We conclude then, that a linear GF(4) code defines a quantum stabilizer code if and only if the code is self-orthogonal in the hermitian inner product. Classical codes with these properties have been much studied.

In particular, consider again the $[5, 3, 3]_4$ Hamming code. Its parity check matrix (in an unconventional presentation) can be expressed as

$$H = \begin{pmatrix} 1 & \omega & \omega & 1 & 0 \\ 0 & 1 & \omega & \omega & 1 \end{pmatrix}, \tag{7.202}$$

which is also the generator matrix of its dual, a linear self-orthogonal $[5, 2, 4]_4$ code. In fact, this $[5, 2, 4]_4$ code, with $4^2 = 16$ codewords, is precisely the stabilizer of the $[[5, 1, 3]]$ quantum code. By identifying $1 \equiv \boldsymbol{X}, \omega \equiv \boldsymbol{Z}$, we recognize the two rows of $H$ as the stabilizer generators $\boldsymbol{M}_1, \boldsymbol{M}_2$. The dual of the Hamming code is a linear code, so linear combinations of the rows are contained in the code. Adding the rows and multiplying by $\omega$ we obtain

$$\omega(1, \bar{\omega}, 0, \bar{\omega}, 1) = (\omega, 1, 0, 1, \omega), \tag{7.203}$$

which is $\boldsymbol{M}_4$. And if we add $\boldsymbol{M}_4$ to $\boldsymbol{M}_2$ and multiply by $\bar{\omega}$, we find

$$\bar{\omega}(\omega, 0, \omega, \bar{\omega}, \bar{\omega}) = (1, 0, 1, \omega, \omega), \tag{7.204}$$

which is $\boldsymbol{M}_3$.

The $[[5, 1, 3]]$ code is just one example of a quite general construction. Consider a subcode $C$ of $GF(4)^n$ that is additive (closed under addition), and self-orthogonal (contained in its dual) with respect to the symplectic inner product. This $GF(4)$ code can be identified with the stabilizer of a binary QECC with length $n$. If the $GF(4)$ code contains $2^{n-k}$ codewords, then the QECC has $k$ encoded qubits. The distance $d$ of the QECC is the minimum weight of a vector in $C^{\perp} \setminus C$.

Another example of a self-orthogonal linear $GF(4)$ code is the dual of the $m = 3$ Hamming code with

$$n = \frac{1}{3}(4^3 - 1) = 21. \tag{7.205}$$

The Hamming code has $4^{n-m}$ codewords, and its dual has $4^m = 2^6$ codewords. We immediately obtain a QECC with parameters

$$[[21, 15, 3]], \tag{7.206}$$

that can correct one error.

## 7.16 Good Quantum Codes

A family of $[[n, k, d]]$ codes is *good* if it contains codes whose "rate" $R = k/n$ and "error probability" $p = t/n$ (where $(t = (d-1)/2)$ both approach a nonzero limit as $n \to \infty$. We can use the stabilizer formalism to prove a "quantum Gilbert-Varshamov" bound that demonstrates the existence of good quantum codes. In fact, good codes can be chosen to be nondegenerate.

The *quantum Gilbert-Varshamov bound* says that

$$|\mathcal{E}^{(2)}| - 1 < 2^{n-k} \left( \frac{1 - 2^{-2n}}{1 - 2^{-2k}} \right). \tag{7.207}$$

is a sufficient condition for the existence of a (possibly degenerate) binary stabilizer code that can correct all Pauli operators in a set $\mathcal{E}$; here $|\mathcal{E}^{(2)}|$ denotes the number of distinct Pauli operators of the form $E_a^\dagger E_b$, where $E_a, E_b \in \mathcal{E}$. One consequence of this bound is that there exist "good" $[[n, k, d = 2t + 1]]$ stabilizer codes that achieve an asymptotic rate $R = k/n = 1 - H_2(2t/n) - (2t/n) \log_2 3$.

Let $\mathcal{S}$ denote the set of all $[[n, k]]$ quantum stabilizer codes. We want to show that this set contains a code that corrects all errors in $\mathcal{E}$. That is, we are seeking a code in $\mathcal{S}$ with stabilizer $S$ such that $S^\perp \setminus S$ contains no element of $\mathcal{E}^{(2)} \setminus \boldsymbol{I}$.

Consider this bipartite graph: Vertices on the left are labeled by codes in $\mathcal{S}$ and vertices on the right are labeled by $n$-qubit Pauli operators, excluding the identity. For each code (with stabilizer $S$), we draw edges connecting that code's vertex to each of its nontrivial logical operators — that is, to each Pauli operator in $S^\perp \setminus S$. Our goal, then, is to show that there is some code which is not connected by an edge to any element of $\mathcal{E}^{(2)} \setminus \boldsymbol{I}$. We'll do that by showing that the total number of edges which connect with elements of $\mathcal{E}^{(2)} \setminus \boldsymbol{I}$ is smaller than the number of elements of $\mathcal{S}$. Thus, once we eliminate from $S$ all the codes that fail to correct $\mathcal{E}$, there must be at least one code left over, which does correct $\mathcal{E}$.

Now, each stabilizer code with $n - k$ generators has $|S^\perp \setminus S| = 2^{n+k} - 2^{n-k}$ nontrivial logical operators; thus there are $2^{n+k} - 2^{n-k}$ edges connecting to each vertex on the left. For each nonzero Pauli operator $x$ on the right, there is an edge connecting it to the code with stabilizer $S$ if and only if $x$ is contained in $S^\perp \setminus S$. Note that the number $N_e$ of such edges does not depend on $x$.

There are two ways to count the total number of edges, which must agree: the number of codes $|\mathcal{S}|$ times the number of edges connecting to each code equals the number of nontrivial Pauli operators times the number of edges connecting to each nontrivial Pauli operator, or

$$|\mathcal{S}| \cdot (2^{n+k} - 2^{n-k}) = (2^{2n} - 1) \cdot N_e \Rightarrow |\mathcal{S}|/N_e = 2^{n-k} \left( \frac{1 - 2^{-2n}}{1 - 2^{-2k}} \right). \tag{7.208}$$

Finally, note that the total number of codes connecting to at least one element of $\mathcal{E}^{(2)} \setminus \boldsymbol{I}$

is no larger than $\left(|\mathcal{E}^{(2)}| - 1\right) \cdot N_e$; therefore, a code which corrects $\mathcal{E}$ (one connecting to no elements of $\mathcal{E}^{(2)} \setminus \boldsymbol{I}$) surely exists provided that

$$|\mathcal{S}| > \left(|\mathcal{E}^{(2)}| - 1\right) \cdot N_e, \tag{7.209}$$

from which eq.(7.327) follows.

*The old argument*

We will only sketch the argument, without carrying out the requisite counting precisely. Let $\mathcal{E} = \{\boldsymbol{E}_a\}$ be a set of errors to be corrected, and denote by $\mathcal{E}^{(2)} = \{\boldsymbol{E}_a^\dagger \boldsymbol{E}_b\}$, the products of pairs of elements of $\mathcal{E}$. Then to construct a nondegenerate code that can correct the errors in $\mathcal{E}$, we must find a set of stabilizer generators such that some generator anti-commutes with each element of $\mathcal{E}^{(2)}$.

To see if a code with length $n$ and $k$ qubits can do the job, begin with the set $\mathcal{S}^{(n-k)}$ of all abelian subgroups of the Pauli group with $n-k$ generators. We will gradually pare away the subgroups that are unsuitable stabilizers for correcting the errors in $\mathcal{E}$, and then see if any are left.

Each nontrivial error $\boldsymbol{E}_a$ commutes with a fraction $\sim 1/2^{n-k}$ of all groups contained in $\mathcal{S}^{(n-k)}$, since it is required to commute with each of the $n-k$ generators of the group. (There is a small correction to this fraction that we may ignore for large $n$.) Each time we add another element to $\mathcal{E}^{(2)}$, a fraction $2^{k-n}$ of all stabilizer candidates must be rejected. When $\mathcal{E}^{(2)}$ has been fully assembled, we have rejected at worst a fraction

$$|\mathcal{E}^{(2)}| \cdot 2^{k-n}, \tag{7.210}$$

of all the subgroups contained in $\mathcal{S}^{(n-k)}$ (where $|\mathcal{E}^{(2)}|$ is the number of elements of $\mathcal{E}^{(2)}$.) As long as this fraction is less than one, a stabilizer that does the job will exist for large $n$.

If we want to correct $t = pn$ errors, then $\mathcal{E}^{(2)}$ contains operators of weight at most $2t$ and we may estimate

$$\log_2 |\mathcal{E}^{(2)}| \lesssim \log_2 \left[ \binom{n}{2pn} 3^{2pn} \right] \sim n \left[ H_2(2p) + 2p \log_2 3 \right]. \tag{7.211}$$

Therefore, nondegenerate quantum stabilizer codes that correct $pn$ errors exist, with asymptotic vote $R = k/n$ given by

$$\log_2 |\mathcal{E}^{(2)}| + k - n < 0, \quad \text{or} \quad R < 1 - H_2(2p) - 2p \log_2 3. \tag{7.212}$$

Thus is the (asymptotic form of the) quantum Gilbert–Varshamov bound.

We conclude that codes with a nonzero rate must exist that protect against errors that occur with any error probability $p < p_{\rm GV} \simeq .0946$. The maximum error probability allowed by the Rains bound is $p = 1/6$, for a code that can protect against every error operator of weight $\leq pn$.

Though good quantum codes exist, the explicit construction of families of good codes is quite another matter. Indeed, no such constructions are known.

## 7.17 Concatenated codes

Up until now, all of the QECC's that we have explicitly constructed have $d = 3$ (or $d = 2$), and so can correct one error (at best). Now we will describe some examples of codes that have higher distance.

A particularly simple way to construct codes that can correct more errors is to concatenate codes that can correct one error. A concatenated code is a code within a code. Suppose we have two $k = 1$ QECC's, an $[[n_1, 1, d_1]]$ code $C_1$ code and an $[[n_2, 1, d_2]]$ code $C_2$. Imagine constructing a length $n_2$ codeword of $C_2$, and expanding the codeword as a coherent superposition of product states, in which each qubit is in one of the states $|0\rangle$ or $|1\rangle$. Now replace each qubit by a length-$n_1$ encoded state using the code $C_1$; that is replace $|0\rangle$ by $|\bar{0}\rangle$ and $|1\rangle$ by $|\bar{1}\rangle$ of $C_1$. The result is a code with length $n = n_1 n_2, k = 1$, and distance no less than $d = d_1 d_2$. We will call $C_2$ the "outer" code and $C_1$ the "inner" code.

In fact, we have already discussed one example of this construction: Shor's 9-qubit code. In that case, the inner code is the three-qubit repetition code with stabilizer generators

$$\boldsymbol{ZZI}\ , \quad \boldsymbol{IZZ}\ , \tag{7.213}$$

and the outer code is the three-qubit "phase code" with stabilizer generators

$$\boldsymbol{XXI}\ , \quad \boldsymbol{IXX} \tag{7.214}$$

(the Hadamard rotated repetition code). We construct the stabilizer of the concatenated code as follows: Acting on each of the three qubits contained in the block of the outer code, we include the two generators $\boldsymbol{Z}_1\boldsymbol{Z}_2, \boldsymbol{Z}_2\boldsymbol{Z}_3$ of the inner code (six generators altogether). Then we add the two generators of the outer code, but with $\boldsymbol{X}, \boldsymbol{Z}$ replaced by the *encoded* operations of the inner code; in this case, these are the two generators

$$\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}\bar{\boldsymbol{I}}, \quad \bar{\boldsymbol{I}}\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}, \tag{7.215}$$

where $\bar{\boldsymbol{I}} = \boldsymbol{III}$ and $\bar{\boldsymbol{X}} = \boldsymbol{XXX}$. You will recognize these as the eight stabilizer generators of Shor's code that we have described earlier. In this case, the inner and outer codes both have distance 1 (*e.g.*, $\boldsymbol{ZII}$ commutes with the stabilizer of the inner code), yet the concatenated code has distance $3 > d_1 d_2 = 1$. This happens because the code has been cleverly constructed so that the weight 1 and 2 encoded operations of the inner code do not commute with the stabilizer of the outer code. (It would have been different if we had concatenated the repetition code with itself rather than with the phase code!)

We can obtain a distance 9 code (capable of correcting four errors) by concatenating the $[[5, 1, 3]]$ code with itself. The length $n = 25$ is the smallest for any known code with $k = 1$ and $d = 9$. (An $[[n, 1, 9]]$ code with $n = 23, 24$ would be consistent with the Rains bound, but it is unknown whether such a code really exists.)

The stabilizer of the $[[25, 1, 9]]$ concatenated code has 24 generators. Of these, 20 are obtained as the four generators $\boldsymbol{M}_{1,2,3,4}$ acting on each of the five subblocks of the outer code, and the remaining four are the *encoded* operators $\bar{\boldsymbol{M}}_{1,2,3,4}$ of the outer code. Notice that the stabilizer contains elements of weight 4 (the stabilizer elements acting on each of the five inner codes); therefore, the code is degenerate. This is typical of concatenated codes.

There is no need to stop at two levels of concatenation; from $L$ QECC's with parameters $[[n_1, 1, d_1]], \ldots, [[n_L, 1, d_L]]$, we can construct a hierarchical code with altogether $L$ levels of codes within codes; it has length

$$n = n_1 n_2 \ldots n_L, \tag{7.216}$$

and distance

$$d \geq d_1 d_2 \dots d_L. \tag{7.217}$$

In particular, by concatenating the $[[5, 1, 3]]$ code $L$ times, we may construct a code with parameters

$$[[5^L, 1, 3^L]]. \tag{7.218}$$

Strictly speaking, this family of codes cannot protect against a number of errors that scales linearly with the length. Rather the ratio of the number $t$ of errors that can be corrected to the length $n$ is

$$\frac{t}{n} \sim \frac{1}{2}\left(\frac{3}{5}\right)^L , \tag{7.219}$$

which tends to zero for large $L$. But the distance $d$ may be a deceptive measure of how well the code performs — it is all right if recovery fails for *some* ways of choosing $t \ll pn$ errors, so long as recovery will be successful for the *typical* ways of choosing $pn$ faulty qubits. In fact, concatenated codes *can* correct $pn$ *typical* errors, for $n$ large and $p > 0$.

Actually, the way concatenated codes are usually used does not fully exploit their power to correct errors. To be concrete, consider the $[[5, 1, 3]]$ code in the case where each of the five qubits is independently subjected to the depolarizing channel with error probability $p$ (that is $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ errors each occur with probability $p/3$). Recovery is sure to succeed if fewer than two errors occur in the block. Therefore, as in §7.5.2, we can bound the failure probability by

$$p_{\text{fail}} \equiv p^{(1)} \leq \binom{5}{2} p^2 = 10p^2. \tag{7.220}$$

Now consider the performance of the concatenated $[[25, 1, 9]]$ code. To keep life easy, we will perform recovery in a simple (but nonoptimal) way: First we perform recovery on each of the five subblocks, measuring $\boldsymbol{M}_{1,2,3,4}$ to obtain an error syndrome for each subblock. After correcting the subblocks, we then measure the stabilizer generators $\bar{\boldsymbol{M}}_{1,2,3,4}$ of the outer code, to obtains its syndrome, and apply an encoded $\bar{\boldsymbol{X}}$, $\bar{\boldsymbol{Y}}$, or $\bar{\boldsymbol{Z}}$ to one of the subblocks if the syndrome reveals an error.

For the outer code, recovery will succeed if at most one of the subblocks is damaged, and the probability $p^{(1)}$ of damage to a subblock is bounded as in eq. (7.220); we conclude that the probability of a botched recovery for the $[[25, 1, 9]]$ code is bounded above by

$$p^{(2)} \leq 10(p^{(1)})^2 \leq 10(10p^2)^2 = 1000p^4. \tag{7.221}$$

Our recovery procedure is clearly not the best possible, because four errors can induce failure if there are two each in two different subblocks. Since the code has distance nine, there is a better procedure that would always recover successfully from four errors, so that $p^{(2)}$ would be of order $p^5$ rather than $p^4$. Still, the suboptimal procedure has the advantage that it is very easily generalized, (and analyzed) if there are many levels of concatenation.

Indeed, if there are $L$ levels of concatenation, we begin recovery at the innermost level and work our way up. Solving the recursion

$$p^{(\ell)} \leq C[p^{(\ell-1)}]^2, \tag{7.222}$$

starting with $p^{(0)} = p$, we conclude that

$$p^{(L)} \leq \frac{1}{C}(Cp)^{2^L}, \tag{7.223}$$

(where here $C = 10$). We see that as long as $p < 1/10$, we can make the failure probability as small as we please by adding enough levels to the code.

We may write

$$p^{(L)} \leq p_o \left(\frac{p}{p_o}\right)^{2^L}, \tag{7.224}$$

where $p_o = \frac{1}{10}$ is an estimate of the *threshold* error probability that can be tolerated (we will obtain better codes and better estimates of this threshold below). Note that to obtain

$$p^{(L)} < \varepsilon, \tag{7.225}$$

we may choose the block size $n = 5^L$ so that

$$n \leq \left\lceil \frac{\log(p_o/\varepsilon)}{\log(p_o/p)} \right\rceil^{\log_2 5}. \tag{7.226}$$

In principle, the concatenated code at a high level could fail with many fewer than $n/10$ errors, but these would have to be distributed in a highly conspiratorial fashion that is quite unlikely for $n$ large.

The concatenated encoding of an unknown quantum state can be carried out level by level. For example to encode $a|0\rangle + b|1\rangle$ in the $[[25, 1, 9]]$ block, we could first prepare the state $a|\bar{0}\rangle + b|\bar{1}\rangle$ in the five qubit block, using the encoding circuit described earlier, and also prepare four five-qubit blocks in the state $|\bar{0}\rangle$. The $a|\bar{0}\rangle + |\bar{1}\rangle$ can be encoded at the next level by executing the encoded circuit yet again, but this time with all gates replaced by encoded gates acting on five-qubit blocks. We will see in the next chapter how these encoded gates are constructed.

## 7.18 Toric code

The toric code is another code family that, like concatenated codes, offers much better performance than would be naively expected based on the code's distance. In addition, the toric code has a further advantage: The check operators (stabilizer generators) that are measured to determine the error syndrome can be chosen to be geometrically local in a two-dimensional layout of the physical qubits. That is, we can measure the check operators using only quantum gates which act on qubits that are near one another. This feature is pleasing to experimentalists building quantum memories, because it is feasible to fabricate devices in which qubits are arranged in a two-dimensional array, and because it can be challenging to realize two-qubit gates acting on qubits that are far apart from one another in space.

To envision the properties of these codes, imagine that the qubits are located on the edges of an $L \times L$ square lattice. We impose periodic boundary conditions, identifying the opposite sides of the square, so that the surface covered by the lattice is topologically equivalent to a doughnut, or torus. This is where the toric code gets its name. It will be convenient to consider this torus topology for describing how the code works, but experimentalists will be relieved to hear that this topology is not really essential; as

we'll explain later, a related code with similar properties can be constructed in which the qubits all reside in the same two-dimensional plane. The planar version of the code has the disadvantage that the qubits on the boundary of the square need to be treated on a different footing than the qubits in the interior, while on the torus, which has no boundary, all qubits can be treated in the same way.

### *7.18.1 Code stabilizer and logical operators*

The toric code is a CSS code, with $\boldsymbol{Z}$-type and $\boldsymbol{X}$-type check operators. Each $\boldsymbol{Z}$-type check operator is associated with a particular *plaquette* of the lattice. I use the term plaquette for an elementary square, or face, of the lattice. A plaquette $P$ contains four edges (also called *links*) of the lattice, and the corresponding check operator $\boldsymbol{Z}_P$ acts on the four qubits which reside on those four edges. If the edges carry labels $1, 2, 3, 4$, then $\boldsymbol{Z}_P = \boldsymbol{Z}_1\boldsymbol{Z}_2\boldsymbol{Z}_3\boldsymbol{Z}_4$; as usual, the four $\boldsymbol{Z}$ operators act on four distinct qubits, but we are suppressing the tensor product symbol to simplify notation. Each $\boldsymbol{X}$-type check operator is associated with a particular vertex (also called a *site*) of the lattice. If four links labeled $1, 2, 3, 4$ meet at the site $s$, then the corresponding check operator is $\boldsymbol{X}_s = \boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3\boldsymbol{X}_4$.

The code space is the simultaneous eigenspace with eigenvalue 1 of all plaquette and site check operators. For this to make sense, the operators must commute, so they can be simultaneously diagonalized. All $\boldsymbol{Z}$-type operators commute with one another trivially, and likewise all $\boldsymbol{X}$-type operators are mutually commuting. It is also true that for any plaquette $P$ and site $s$, the operators $\boldsymbol{Z}_P$ and $\boldsymbol{X}_s$ commute. Either the two operators have disjoint support, in which case they obviously commute, or else the site $s$ lies at one of the corners of the plaquette $P$, in which case the two operators "collide" at two edges. In the latter case, the operators commute because two minus signs arise when $\boldsymbol{Z}_P$ moves past $\boldsymbol{X}_s$.

Our $L \times L$ lattice on the torus has $L^2$ sites, $L^2$ plaquettes, and $2L^2$ edges (there are $L^2$ vertically oriented edges and $L^2$ horizontally oriented edges). Therefore, the code has length $n = 2L^2$. To find the number $k$ of encoded qubits, we need to count the number of *independent* generators of the code stabilizer group. In fact, because every link is shared by two plaquettes, we see that $\prod_P \boldsymbol{Z}_P = \boldsymbol{I}$, where the product is over all lattice plaquettes. There are no further constraints, so the number of independent $\boldsymbol{Z}$-type check operators is $L^2-1$. Likewise, because each link connects two sites, we have $\prod_s \boldsymbol{X}_s = \boldsymbol{I}$, so we conclude that there are $L^2-1$ independent $\boldsymbol{X}$-type check operators. Hence we find that the number of encoded qubits is $2L^2 - 2(L^2 - 1) = 2$.

Recall that the logical operators of a stabilizer code are Pauli operators which commute with the code stabilizer but are not in the stabilzer. In the CSS case, the code has a $\boldsymbol{Z}$-type stabilizer $S_Z$ and an $\boldsymbol{X}$-type stabilizer $S_X$; the $\boldsymbol{Z}$-type logical operators lie in the normalizer $(S_X)^\perp$ of $S_X$ but are not contained in $S_Z$. To understand the logical operators of the code, it will be helpful to have handy characterizations of $(S_X)^\perp$ and $S_Z$.

An arbitrary element of $S_Z$ is a product of plaquette operators, which may be expressed as

$$\tilde{\boldsymbol{Z}}(\omega) = \prod_P \boldsymbol{Z}_P^{\omega_P}. \tag{7.227}$$

Here $\omega_P$ is a binary string of length $L^2$, where $\boldsymbol{Z}_P$ is included in the product for $\omega_P = 1$

and not included for $\omega_P = 0$. (This way of representing $\boldsymbol{Z}(\omega)$ is slightly redundant because the $L^2$ plaquette operators are not quite indenpendent — they satisfy one relation — but that redudancy will not cause any trouble.) Note that we may regard $\omega$ as an element of a vector space over $\mathbb{F}_2$, where addition of vectors corresponds to multiplication of $\boldsymbol{Z}$-type stabilizer operators:

$$\tilde{\boldsymbol{Z}}(\omega)\tilde{\boldsymbol{Z}}(\omega') = \tilde{\boldsymbol{Z}}(\omega + \omega'). \tag{7.228}$$

We denote this vector space by $V_2$, where the subscript 2 signifies that a bit is assigned to each 2-cell (that is, plaquette) of the lattice. A vector $\omega$ in $V_2$ is called a *2-chain*.

Recall that any $\boldsymbol{Z}$-type operator can be expressed as a binary string of length $n = 2L^2$, which we denote $\epsilon$:

$$\boldsymbol{Z}(\epsilon) = \bigotimes_{\ell} \boldsymbol{Z}_{\ell}^{\epsilon_\ell}, \tag{7.229}$$

where the tensor product is over all lattice edges. Here, too, we may regard $\epsilon$ as an element of a binary vector space, which we denote $V_1$ because a bit is assigned to each 1-cell (that is, edge) of the lattice. A vector in $V_1$ is called a *1-chain*, and addition of 1-chains corresponds to multiplication of $\boldsymbol{Z}$-type operators.

There is a linear operator $\partial_2$ which maps $V_2$ to $V_1$, called the boundary operator. Under this map the bit assigned to the edge $\ell$ by the 1-chain $\partial_2\omega$ is the mod-2 sum of the bits assigned by the 2-chain $\omega$ to the two plaquettes which contain the edge $\ell$. It follows that

$$\tilde{\boldsymbol{Z}}(\omega) = \boldsymbol{Z}(\partial_2\omega). \tag{7.230}$$

The boundary operator is aptly named. If a stabilizer element $\tilde{\boldsymbol{Z}}(\omega)$ is a product of plaquette check operators which fill a connected "droplet" on the lattice, then $\tilde{\boldsymbol{Z}}(\omega)$ is supported on the outer edge (boundary) of this droplet. We have now obtained the handy characterization of $S_Z$ that we wanted: a $\boldsymbol{Z}$-type operator $\boldsymbol{Z}(\epsilon)$ is in $S_Z$ if and only if the 1-chain $\epsilon$ is the boundary of some 2-chain $\omega$ ($\epsilon = \partial_2\omega$). In other words, $S_Z$ is the *image* of the map $\partial_2$.

Now consider the $\boldsymbol{Z}$-type Pauli operators in the normalizer $(S_X)^\perp$, which commute with all the site-operators $\{\boldsymbol{X}_s\}$. For the purpose of characterizing these operators, we introduce another vector space, denoted $V_0$, because each vector in $V_0$ (called a *0-chain*) assigns a bit to each 0-cell (that is, site) of the lattice. Another boundary operator, denoted $\partial_1$, may be defined, a linear operator mapping $V_1$ to $V_0$. Under this mapping, the 0-chain $\partial_1\epsilon$ assigns to a site $s$ the mod-2 sum of the bits assigned by the 1-chain $\epsilon$ to the four edges which meet at $s$. It follows that the $\boldsymbol{Z}$-type operator $\boldsymbol{Z}(\epsilon)$ commutes with the $\boldsymbol{X}$-type check operator if and only if $(\partial_1\epsilon)_s = 0$. Therefore, $\boldsymbol{Z}(\epsilon)$ is contained in $(S_X)^\perp$ (commutes with the stabilizer) if and only if $\partial_1\epsilon = 0$. A 1-chain with trivial boundary is said to be a *1-cycle*. Note the property $\partial_1 \circ \partial_2 = 0$ (the boundary of a boundary is zero) which just means that elements of the $\boldsymbol{Z}$-type stabilizer commute with the $\boldsymbol{X}$-type stabilizer.

The vector spaces $V_0, V_1, V_2$, together with the boundary operators $\partial_1, \partial_2$ (where $\partial_1 \circ \partial_2 = 0$) constitute what is called a *chain complex*:

$$V_2 \xrightarrow{\partial_2} V_1 \xrightarrow{\partial_1} V_0. \tag{7.231}$$

This chain complex provides a handy way to describe the code's logical Pauli operators: $\boldsymbol{Z}(\epsilon)$ is a logical operator (which might be trivial) iff $\epsilon$ is a 1-cycle (in the *kernel* of $\partial_1$).

Since the trivial logical operators are those in the image of $\partial_2$, we see that the algebra $\mathcal{L}_Z$ of the $\boldsymbol{Z}$-type logical operators is

$$\mathcal{L}_Z = \frac{\mathrm{Ker}(\partial_1)}{\mathrm{Im}(\partial_2)}. \tag{7.232}$$

The code's $\boldsymbol{X}$-type logical operators have exactly the same structure. This is most easily seen by invoking the concept of a *dual lattice*. The dual of a square lattice is another square lattice, which is obtained from the primal lattice by replacing each plaquette of the primal lattice by a site of the dual lattice, each edge of the primal lattice by a 90° rotated edge of the dual lattice, and each site of the primal lattice by a plaquette of the dual lattice. Under duality, then, the $\boldsymbol{X}$-type check operators on sites of the primal lattice are transformed to plaquette operators on the dual lattice, and the $\boldsymbol{Z}$-type check operators on plaquettes of the primal lattice are transformed to site operators of the dual lattice. Therefore, the logical operators are associated with the same chain complex as before, but with $\boldsymbol{Z}$ and $\boldsymbol{X}$ interchanged.

Something more about the stringlike operators, cycles of the torus, etc.

### *7.18.2 Chain complex for a general CSS code*

The chain complex concept provides a useful way of thinking about logical operators not just for the toric code, but for any CSS stabilizer code. Recall that a CSS code is defined by two check matrices — $H_Z$ and $H_X$, each with $n$ columns. The $m_Z$ rows of $H_Z$ correspond to the code's $m_Z$ independent $\boldsymbol{Z}$-type stabilizer generators, and the $m_X$ rows of $H_X$ correspond to the code's $m_X$ independent $\boldsymbol{X}$-type stabilizer generators.

Consider the $\boldsymbol{Z}$-type logical operators. A general element of the $\boldsymbol{Z}$-type stabilizer can be expressed as $\tilde{\boldsymbol{Z}}(\omega) = \boldsymbol{Z}(\epsilon)$. Here $\omega$, a vector of length $m_Z$ specifying which stabilizer generators are multiplied together, may be regarded as a 2-chain, while $\epsilon$, a vector of length $n$ which specifies the support of the stabilizer element, may be regarded as a 1-chain, related to $\omega$ by $\epsilon = \partial_2\omega$. The boundary operator $\partial_2$ is defined as

$$\partial_2\omega = H_Z^T\omega. \tag{7.233}$$

Any $\boldsymbol{Z}$-type operator $\boldsymbol{Z}(\epsilon)$ commutes with all $\boldsymbol{X}$-type stabilizer generators iff $\partial_1\epsilon = 0$, where

$$\partial_1\epsilon = H_X\epsilon; \tag{7.234}$$

the length-$m_X$ vector $\partial_1\epsilon$ may be regarded as a 0-cell. Hence the $\boldsymbol{Z}$-type logical algebra $\mathcal{L}_Z$ corresponds to the coset space

$$\mathcal{L}_Z = \frac{\mathrm{Ker}(\partial_1)}{\mathrm{Im}(\partial_2)} = \frac{\mathrm{Ker}(H_X)}{\mathrm{Im}(H_Z^T)}. \tag{7.235}$$

The property

$$\partial_1 \circ \partial_2 = H_X H_Z^T = 0 \tag{7.236}$$

ensures that the $\boldsymbol{X}$-type and $\boldsymbol{Z}$-type stabilizer generators commute. The $\boldsymbol{X}$-type logical operators can be described similarly, but with $H_X$ and $H_Z$ interchanged, so that

$$\mathcal{L}_X = \frac{\mathrm{Ker}(H_Z)}{\mathrm{Im}(H_X^T)}. \tag{7.237}$$

To make contact with our earlier discussion of CSS codes, note that the rows of $H_X$

span the classical code $C_2 = \mathrm{Im}(H_X^T)$, and that $\mathrm{Ker}(H_Z) = C_1$ is the classical code which satisfies the $H_Z$ parity check. The property $H_Z H_X^T = 0$ means that $C_2$ is a subcode of $C_1$, and we find

$$\mathcal{L}_X = C_1/C_2, \quad \mathcal{L}_Z = C_2^\perp/C_1^\perp. \tag{7.238}$$

This matches up with our earlier description of the logical operators, in which the code is spanned by "coset states," each a uniform superposition of representatives of a coset of $C_2$ in $C_1$, or, in the conjugate basis of $C_1^\perp$ in $C_2^\perp$. [???]

### 7.18.3 Error recovery

As for any CSS code, we may perform correction of $\boldsymbol{Z}$ errors and correction of $\boldsymbol{X}$ errors separately for the toric code. Let's begin by considering the correction of $\boldsymbol{Z}$ errors. Suppose that an unknown initial quantum state $|\psi\rangle$ in the code space is prepared, and then $\boldsymbol{Z}$ errors occur on some of the edges of the lattice. After the errors occur, we measure all the site check operators $\{\boldsymbol{X}_s\}$, finding an eigenvalue which is either $+1$ or $-1$ at each site $s$. These measurement outcomes constitute the syndrome of the error, providing partial information about where the $\boldsymbol{Z}$ error occurred. Our job is the infer from this syndrome a recovery operation which has a good chance of successfully reversing the damage and recovering the initial state $|\psi\rangle$.

In the chain complex language, the unknown error $\boldsymbol{Z}(\epsilon)$ is described by a 1-chain $\epsilon$, and measuring the check operators determines its boundary $\sigma = \partial_1 \epsilon$. To attempt recovery, we return the state to the code space by applying the operator $\boldsymbol{Z}(\varphi)$, chosen so that $\boldsymbol{Z}(\varphi)\boldsymbol{Z}(\epsilon) = \boldsymbol{Z}(\varphi + \epsilon)$ commutes with code stabilizer; that is, $\partial_1(\varphi + \epsilon)$. Our attempt succeeds if $\boldsymbol{Z}(\varphi + \epsilon)$ is in the code stabilizer, or $\varphi + \epsilon \in \mathrm{Im}(\partial_2)$. Otherwise $\boldsymbol{Z}(\varphi + \epsilon)$ is a nontrivial logical operator and our attempt fails.

How $\varphi$ should be chosen depends on the noise model governing the $\boldsymbol{Z}$-type errors. To be concrete, consider a model in which the $\boldsymbol{Z}$ errors are independent and identically distributed (i.i.d.): each qubit either suffers a $\boldsymbol{Z}$ error, with probability $\varepsilon$, or is undamaged, with probability $1 - \varepsilon$. We will choose $\varphi$ so that $\boldsymbol{Z}(\varphi)$ is the error of lowest weight which produces the observed syndrome $\sigma$, because the error of lowest weight is the most likely error when the noise is i.i.d. (with $\varepsilon < 1/2$).

We say that the recovery procedure is *optimal* if it achieves the lowest possible probability of failure. For a subtle reason, our minimal-weight criterion is not necessarily the optimal method. The elements of $\mathrm{Ker}(\partial_1)/\mathrm{Im}(\partial_2)$ are called *homology classes*, and the optimal procedure is to choose the 1-chain $\varphi$ so that [???]. The subtlety is that the most likely cycle does not necessarily lie in the most likely homology class. Nevertheless, choosing $\varphi$ of minimal weight is nearly optimal and good enough for our purposes. Fortunately, there is an efficient classical algorithm which finds the minimal 1-chain with given boundary $\sigma$, so the

### 7.18.4 Planar surface codes

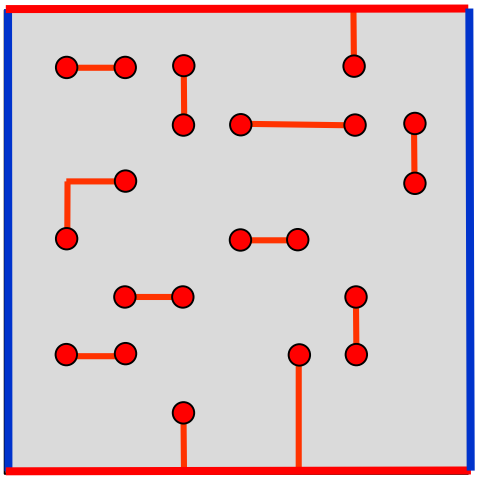
### 7.18.5 Surface code accuracy threshold

To create a stable quantum memory, we need not synthesize a topologically ordered material; instead we can *simulate* the material using whatever quantum computing hardware we prefer. Kitaev constructed a simple two-dimensional lattice model (the *surface*

*code*), with a qubit at each lattice site, that exhibits $\mathbb{Z}_2$ topological order. Though it was first proposed 25 years ago, the surface code still offers a particularly promising route toward scalable fault-tolerant quantum computation. It has two major advantages. First, the quantum processing needed to diagnose and correct errors is remarkably simple. Second, and not unrelatedly, it can tolerate a relatively high gate error rate.

Errors afflicting a quantum memory can be expanded in terms of multi-qubit Pauli operators, and each such Pauli operator can be expressed as a product of an $X$-type error, where either $X$ or the identity acts on each qubit, and a $Z$-type error, where either $Z$ or the identity acts on each qubit. (A $Y = -iZX$ error is just the case where both $X$ and $Z$ act on the same qubit.) Therefore, our quantum memory will be well protected if we can correct both $X$-type and $Z$-type errors with high success probability. In the case of the surface code, there are two separate procedures for correcting $X$ errors and correcting $Z$ errors, and both work in essentially the same way, so it will suffice to discuss only how the $Z$ errors are corrected.

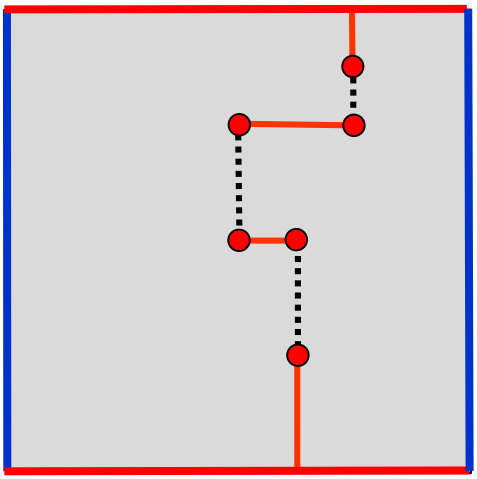
In (one version of) the surface code, the physical qubits reside on edges of a square lattice, and $e$ anyons may reside on the sites of the lattice. Suppose an unknown quantum state $\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$ has been stored in the code space, where $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are the encoded $\bar{Z}$ eigenstates. After this state is encoded, $Z$ errors occur on some of the qubits, knocking the state out of the code space by creating $e$ anyons. A snapshot of a typical error configuration is shown here:



Edges on which the qubits have $Z$ errors (colored red) define a set of connected "error chains," and pairs of anyons appear at the endpoints of each error chain. The positions of the anyons (and hence the endpoints of the error chains) can be identified by a simple quantum computation. After finding their positions we can remove these anyons two at a time; we select a pair of anyons, and apply $Z$ to all the qubits along a "recovery chain" that connects the pair, in effect bringing the pair of anyons together to annihilate. Alternatively, we can remove a single anyon by choosing a recovery chain that connects that anyon to the top or bottom edge. Our goal is to remove all of the anyons, returning the state to the code space, and (we hope) restoring the initial encoded state.

The anyon positions are said to constitute an error "syndrome" because they help us to diagnose the damage sustained by the physical qubits in the code block. Even though this syndrome locates the boundary points of the error chains, we don't know the configuration of the error chains themselves, so our recovery chains won't necessarily

coincide with the error chains, or even connect together the same pairs of anyons. But they don't have to. If each connected path resulting from combining the error chain with the recovery chain forms a closed loop in the bulk, or an open path with both its endpoints lying on the same edge (either top or bottom), then error recovery is successful. This works because of the properties of the topologically ordered medium: creation of a pair of anyons followed by pair annihilation, or creation of a single anyon at the bottom (top) edge followed by annihilation at the bottom (top) edge are processes that act trivially on the code space. On the other hand, suppose that the error chain combined with the recovery chain produces a path connecting the bottom and top edges as shown here:



Then (if there are an odd number of such paths) a logical $\bar{Z}$ error occurs and our recovery procedure fails.

To keep things simple, consider a stochastic independent noise model, in which each qubit in the code block experiences a $Z$ error with probability $\epsilon$. Suppose we choose our recovery chains to have the minimal possible weight; that is, we return to the code space by applying $Z$ to as few qubits as possible. Given the known positions of the anyons, this minimal chain can be computed efficiently with a classical computer. For this recovery procedure, we can find an upper bound on the probability of a logical error by the following argument.

We denote by $d$ the minimal weight of a connected path from the bottom edge to the top edge; that is, $d$ (the *distance* of the code) is the minimal weight of a $\bar{Z}$ logical operator. If our attempt to recover resulted in a logical $\bar{Z}$ error, there must be a path connecting the bottom and top edges of the code block such that each edge of the lattice on this path is in either an error chain or a recovery chain. Let's say this connected path has length $\ell \geq d$, and denote the path by $C_\ell$. The number of errors on $C_\ell$ must be at least $\ell/2$ if $\ell$ is even, or $(\ell+1)/2$ if $\ell$ is odd; otherwise we could have found a lower weight recovery chain by applying $Z$ on the error chains contained in $C_\ell$, rather than to the qubits on $C_\ell$ which are complementary to the error chains on $C_\ell$. The number of ways that the edges with $Z$ errors could be distributed along $C_\ell$ is no more than $2^\ell$ (each qubit on $C_\ell$ either has an error or does not). Since, for each physical qubit, $Z$ errors occur with probability $\epsilon$, the probability that $C_\ell$ is contained in the union of error chains and recovery chains obeys

$$P(C_\ell) \leq 2^\ell \epsilon^{\ell/2}. \tag{7.239}$$

Let $N_\ell$ denote the number of paths connecting the bottom and top edges with length $\ell$. For a logical error to occur, the combination of error chains and recovery chains must produce at least one path connecting the bottom and top edges. Using the upper bound eq.(7.239) on the probability of each such path, and applying the union bound, we conclude that the probability of a logical $\bar{Z}$ error satisfies

$$P_{\text{logical}} \leq \sum_{\ell=d}^{n} N_\ell 2^\ell \epsilon^{\ell/2}. \tag{7.240}$$

The lower limit on the sum is $\ell = d$, the length of the shortest path connecting the bottom and top edges. The upper limit is $n$, the total number of qubits in the code block, which is therefore the maximum length of any path.

We can also find a simple upper bound on $N_\ell$. Let's say our square lattice is $d \times d$. A path from the bottom to the top edge can begin at any one of $d$ positions along the bottom edge, and in each of the $\ell$ steps along the path, there are three possible moves: straight ahead, left turn, or right turn. Therefore (even if we don't insist that the path reach the top edge), we have

$$N_\ell \leq d\ 3^\ell \implies P_{\text{logical}} \leq d \sum_{\ell=d}^{n} (36\epsilon)^{\ell/2}\,. \tag{7.241}$$

Now suppose that $\epsilon < 1/36$, so that the terms in the sum over $\ell$ decrease as $\ell$ increases. For a square lattice, the number of edges (qubits) in the code block is $n = O(d^2)$, so the number of terms in the sum is also $O(d^2)$, and we conclude that

$$P_{\text{logical}} \leq O(d^3)\,(\epsilon/\epsilon_0)^{d/2} \quad \text{for} \quad \epsilon < \epsilon_0 = 1/36 \approx .028\ . \tag{7.242}$$

Thus, this argument establishes that the surface code is a quantum memory with an *accuracy threshold* — for any constant $\epsilon < \epsilon_0$, the probability of a logical error decays exponentially as the code distance $d$ increases (apart from a possible polynomial prefactor). If the physical error rate is below the threshold value $\epsilon_0$, we can make the logical error rate arbitrarily small by choosing a sufficiently large code block. Unsurprisingly, in view of the crudeness of this argument, the actual value of the error threshold $\epsilon_0$ is larger than we estimated. Monte Carlo simulations find $\epsilon_0 \approx .103$ for this minimum-weight decoding method.

### *7.18.6 Scalable quantum computing*

To draw quantitative conclusions about the overhead cost of fault-tolerant quantum computing, refinements of this argument are needed. First of all, we implicitly assumed that the error syndrome measurements are perfect. In fact measurement errors occur, which means we need to repeat the measurement $O(d)$ times to acquire sufficiently trustworthy information about where the anyons are located. Secondly, we did not take into account the structure of the quantum circuit used to make these measurements. To determine whether an anyon is present at a particular site, four entangling two-qubit gates are needed, any one of which could be faulty, and a single fault can cause both an error in the measurement outcome and errors in the data qubits. A more complete analysis shows that the threshold error rate for the two-qubit gates is close to 1%. Numerical simulations find that for each round of syndrome measurement, the probability

of a logical error rate scales roughly like

$$P_{\text{logical}} \approx 0.1\,(100p)^{(d+1)/2}\,, \tag{7.243}$$

where now $p$ denotes the two-qubit gate error rate, and we have assumed that $d$ is odd.

So far we have considered only the probability of a storage error for one protected qubit, but in a scalable fault-tolerant quantum computer we will need many protected qubits, and we will need to perform highly reliable universal quantum gates that act on these qubits. One can envision an architecture in which the logical qubits are arranged like square tiles on a surface, with buffer qubits filling gaps between the tiles. I won't go into the details here of how the logical gates are executed, but it is helpful to realize that much of the logical processing can be executed by performing entangling measurements on pairs of logical blocks. For example, we can measure $\bar{X}_1 \otimes \bar{X}_2$, where blocks 1 and 2 reside on adjacent tiles, by fusing the blocks together along their red edges and then cutting the blocks apart again, a process called "lattice surgery." The fusing and cutting are achieved by measurements that activate the buffer qubits in between the edges of the two blocks, followed by measurements that decouple the buffer qubits.

The good news is that the error rates for logical gates are not much worse than the storage error rates we have already discussed, except we should keep in mind that we need to repeat the syndrome measurement $O(d)$ times in each logical gate cycle. The bad news is that eq.(7.243) indicates that we'll need a rather large code distance if we want to make the logical error rate very small. Suppose, for example, that we would like to run Shor's algorithm to factor a 2048-bit number, which would break the RSA cryptosystem, and suppose that the physical two-qubit gate error rate is $10^{-3}$, better than in current multi-qubit devices. A recent analysis calls for a logical error probability $\approx 10^{-15}$ per round of syndrome measurement, and hence a code distance of $d = 27$. The number of physical qubits per code block, including ancilla qubits needed for syndrome measurement and lattice surgery, is $2(d+1)^2 = 1568$, and the total number of logical qubits used in this version of the factoring algorithm is about 14,000, pushing the physical qubit count above 20 million. That's a lot!

There are many challenges to making large-scale fault-tolerant quantum computing practical, including serious systems engineering issues. There are also issues of principle to consider — what is required for a fault-tolerant scheme to be scalable, and what conditions must be satisfied by the noise model? One essential requirement is some form of cooling, to extract the entropy introduced by noise. In the protocol described above, entropy is extracted by measuring and resetting ancilla qubits in each round of syndrome measurement. Parallel operations are also necessary, so noise can be controlled in different parts of the computer simultaneously.

The analysis leading to eq.(7.243) is based on a simple noise model in which gate errors are stochastic (rather than coherent) and there are no correlations among errors in different gates. The fault-tolerant methods should work for more realistic noise models, as long as the errors are sufficiently weak and not too strongly correlated. By benchmarking logical error rates using relatively small quantum codes in near-term devices, we will gain valuable insight into how effectively quantum error correction protects computations performed on actual quantum hardware.

## 7.19 A bound on topological stabilizer codes

We say that a stabilizer code family is *local* if we may choose all stabilizer generators to be geometrically local in $D$ spatial dimensions, for some finite $D$. If in addition the code distance $d$ increases without bound as the code length $n$ increases, we say that the code is *topological*. For example, if the qubits are arranged in a $D$-dimensional (hyper)cubic lattice, the code is local if each stabilizer generator has its support on a (hyper)cube with side length $r$, where $r$ is a constant independent of the length $n$ of the code. When a code is local in $D$ dimensions, we may call it a *$D$-dimensional code*, dropping the word "local" which is understood. We say that $r$ is the *range* of the code generators. Note that for a topological code family, any set of qubits of constant size is correctable for sufficiently large $n$.

Here we prove a bound on the code parameters $[[n, k, d]]$ for two-dimensional (2D) stabilizer codes: $kd^2 = O(n)$.

The proof of the bound makes use of three lemmas — the cleaning lemma, the union lemma, and the expansion lemma. The cleaning lemma for stabilizer codes asserts that for any correctable set $M$ and logical Pauli operator $x$, we may choose $x$ to be supported on the complement $M^c$ of $M$. That is, for any logical operator $x$ (commuting with the code stabilizer $S$) there exists $y \in S$ such that $xy$ is supported on $M^c$. A proof of the cleaning lemma is provided in the previous section.

### *7.19.1 The union lemma*

We say that two sets of qubits are *separated* if no stabilizer generator has support on both sets. For a 2D code this is ensured if no $r \times r$ square intersects with both sets. The union lemma says that if $M_1$ and $M_2$ are separated and both are correctable, then their union $M_1 \cup M_2$ is also correctable. We denote this union as $M_1 M_2$. To prove the union lemma by contradiction, suppose that $M_1 M_2$ is not correctable, in which case there exists a nontrivial logical operator $x$ supported on $M_1 M_2$, of the form

$$x = (y_1)_{M_1} \otimes (y_2)_{M_2} \otimes I_{(M_1 M_2)^c}. \tag{7.244}$$

Because $x$ is logical, it commutes will all stabilizer generators, and since no generator has support on both $M_1$ and $M_2$, it must be that

$$x_1 = (y_1)_{M_1} \otimes I_{M_2} \otimes I_{(M_1 M_2)^c} \quad \text{and} \quad x_2 = I_{M_1} \otimes (y_2)_{M_2} \otimes I_{(M_1 M_2)^c} \tag{7.245}$$

also commute with all stabilizer generators, and hence are logical. Furthermore, since $x = x_1 x_2$, and $x$ is nontrivial, either $x_1$ or $x_2$ must be nontrivial. This contradicts our assumption that $M_1$ and $M_2$ are correctable, proving the lemma.

An alternative way to formulate the proof is to note that if $x$ is any logical operator, then we can clean either $M_1$ or $M_2$, because both are correctable. But because $M_1$ and $M_2$ are separated, the stabilizer generators that clean $M_1$ do not act on $M_2$, and the stabilizer generators that clean $M_2$ do not act on $M_1$. Therefore we can clean $M_1$ and $M_2$ simultaneously; i.e., we can clean the union $M_1 M_2$. This means that all logical operators can be supported on $(M_1 M_2)^c$, $g((M_1 M_2)^c) = 2k$, and hence by the cleaning lemma $g(M_1 M_2) = 0$. We conclude that $M_1 M_2$ supports no nontrival logical operators and is therefore correctable.

### 7.19.2 The expansion lemma

The expansion lemma requires a bit more explanation. Let $M'$ denote the support of all stabilizer generators which act nontrivially on $M$. The *external boundary* $\partial_+ M$ of $M$ is $M' \cap M^c$, and the *internal boundary* $\partial_- M$ of $M$ is $(M^c)' \cap M$. For a local code with range $r$, we may visualize $\partial_+ M$ as a thin shell surrounding $M$ (with thickness no greater than $r$), while $\partial_- M$ is a thin shell surrounding $M^c$. The expansion lemma specifies conditions under which we can slightly augment the size of a correctable set while preserving its correctability. It asserts the following: Suppose that $M = NA$ (that is, $M = N \cup A$, where $N$ and $A$ are disjoint). Suppose further that $A$ contains $\partial_- M$, and that $N$, $A$, and $\partial_+ M$ are correctable. Then $M$ is correctable.

To prove the expansion lemma by contradiction, suppose that $M$ is not correctable, in which case there is a nontrivial logical Pauli operator $x$ which is supported on $M$. Furthermore, because $A$ is correctable, there is a stabilizer element $y$, which we may choose to be supported on $M'$, such that $z = xy$ is cleaned on $A$:

$$z = w_N \otimes I_A \otimes v_{\partial_+ M} \otimes I_{(M')^c}. \tag{7.246}$$

Because $z$ is logical, it commutes with all stabilizer generators, and because $A \supseteq \partial_- M$, no stabilizer generator acts nontrivially on both $N = M \setminus A$ and $\partial_+ M$. Therefore, the restriction $z_1$ of $z$ to $N$ is logical, and the restriction $z_2$ of $z$ to $\partial_+ M$ is also logical. Furthermore, since $z = z_1 z_2$ and $z$ is nontrivial, either $z_1$ or $z_2$ must be nontrivial. This contradicts our assumption that $N$ and $\partial_+ M$ are correctable, proving the lemma.

### 7.19.3 The holographic principle

Next, we use the cleaning lemma and expansion lemma to infer the *holographic principle* for 2D codes, which asserts that, for some constant $\alpha$, any square on the lattice with side length $\ell = \alpha d$ is a correctable set of qubits. Note that this is not obvious, since the number of qubits contained in the square is $\Omega(d^2)$, and hence, for a topological code family becomes much larger than the code distance $d$ (the size of the smallest noncorrectable set) for sufficiently large $n$.

To prove the holographic principle, consider two concentric squares $N$ and $M$, where $N$ has side length $\ell - 4r$, and $M$ has side length $\ell - 2r$; therefore, $M'$ is contained in a square with side length $\ell$, and $A = M \setminus N$ contains $\partial_- M$. The expansion lemma ensures that $M$ is correctable if $N$, $A$, and $\partial_+ M$ are all correctable. By definition of distance a set is correctable if it contains fewer than $d$ qubits, and therefore $\partial_+ M$ and $A$ are both correctable provided that

$$|\partial_+ M| \leq |M'| - |M| \leq \ell^2 - (\ell - 2r)^2 < 4r\ell < d. \tag{7.247}$$

Now imagine starting with a small square, with area less than $d$ so that the square is correctable, and gradually increasing the side length step-by-step, where the length grows by $2r$ in each step. According to eq.(7.247), the square continues to be correctable as long as the side length $\ell$ remains less than $d/4r$. This proves the holographic principle. We call this the holographic principle, somewhat whimsically, because if logical information is encoded in the square, then some of that logical information (though perhaps not all) must be accessible near the square's boundary.

### 7.19.4 Applying the decoupling principle

To prove our bound on $[[n, k, d]]$ we need one more result, which applies to more general binary quantum codes, not just to stabilizer codes. Suppose the code block is divided into three regions $ABC$, where $A$ and $B$ are both correctable. Then the number $k$ of encoded qubits can be no larger than $|C|$, the number of qubits in $C$.

We obtain this result from the decoupling principle. Express the code block as $MM^c$, where $M$ is a correctable set and $M^c$ is its complement. Introduce a reference system $R$, of dimension $2^k$, which is maximally entangled with the code's $k$ logical qubits. Consider the Stinespring dilation of the erasure channel applied to $M$. The dilation discards $M$, mapping it to the environment $E$; hence we may consider $M$ itself to be the environment of the channel. The decoupling principle asserts that if the erasure channel is correctable, then the marginal density operator for $RM$ is a product state, whose entropy is additive — the reference system is uncorrelated with the erased subsystem:

$$H(RM) = H(R) + H(M) \implies k = H(M^c) - H(M), \tag{7.248}$$

where in the second equation we have replaced the entropy $H(R)$ of the maximally mixed reference system by its number of qubits $k$, and have used the feature that the state of $RMM^C$ is pure, hence $H(RM) = H(M^c)$.

Now, recalling that $A$ is correctable we apply eq.(7.248) to $M = A$ and its complement $M^c = BC$, obtaining

$$k = H(BC) - H(A) \leq H(B) + H(C) - H(A). \tag{7.249}$$

Likewise, since $B$ is also correctable, we apply eq.(7.248) to $M = B$ and its complement $M^c = AC$, obtaining

$$k = H(AC) - H(B) \leq H(A) + H(C) - H(B). \tag{7.250}$$

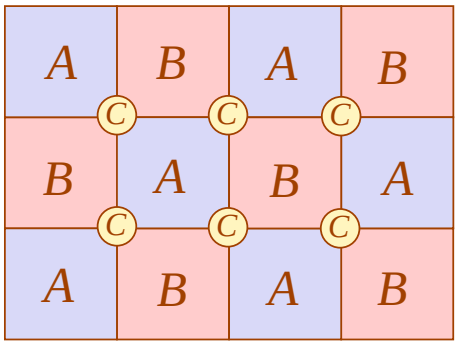
Combining these two inequalities we find

$$k \leq H(C) \leq |C|, \tag{7.251}$$

as desired.

### 7.19.5 Proof of the bound

Now we are ready to prove our bound on 2D topological codes. For a 2D stabilizer code, consider partitioning the code block into three sets $ABC$ as shown here:



Each connected component of $A$ is a square with side length less than $d/4r$, with corners clipped off as shown, and same for $B$. Each connected component of $C$ has constant size, but is sufficiently large to separate the components of $A$ by a distance larger than the range $r$ of the stabilizer generators (and same for $B$). By the holographic principle, each component of $A$ is correctable, and because the components are separated, $A$ is correctable by the union lemma. Likewise, $B$ is also correctable. The number of components of $C$ is $O(n/d^2)$, and each component has constant size. We conclude that

$$k \leq |C| = O(n/d^2), \tag{7.252}$$

which proves the bound. Though in the figure we have not indicated any holes punched through the code block, adding holes would not modify the conclusion.

We can also see that the bound is tight — 2D code families exist with $kd^2$ linear in $n$. To show this, consider a planar surface code defined on a rectangle with $k$ holes punched through it, and with boundary conditions such that $e$ anyons can "condense" (i.e., disappear) at any boundary. Nontrivial logical operators are $e$ strings that connect distinct boundaries, and $m$ strings that enclose one or more holes. The code distance (the smallest weight of a nontrivial logical operator) will be at least $d$ if each hole has a perimeter at least $d$, and no two boundaries are closer together than distance $d$. We can easily arrange $k$ square holes, each with side length $d/4$ such that all holes are distance $d$ apart, and also distance $d$ from the outer boundary, on a rectangular lattice $2d$ sites high and $2kd$ sites wide, hence with $4kd^2$ sites; the corresponding surface code has $n = O(kd^2)$ physical qubits.

Our bound shows that a 2D code family cannot be "good" — it cannot have both $k$ and $d$ scaling linearly with $n$. In fact, if $k$ scales linearly with $n$, then the distance $d$ must be a constant. To do better, we need to either allow non-Euclidean geometry, or to allow stabilier generators that are not geometrically local.

## 7.20 String operators and self-correcting codes

### *7.20.1 2D stabilizer codes have stringlike logical operators*

Using the cleaning lemma and the union lemma we can see that any 2D stabilizer code has a nontrivial logical operator supported on a strip of constant width. A similar argument shows that a stabilizer code in $D$ simensions has a nontrivial logical operator supported on a $(D-1)$-dimensional slab of constant thickness. (Note that the argument does not show that each logical operator is supported on a slab — only that at least one nontrivial logical operator has this structure.)

For qubits arranged on a square, we may argue as follows. Cover the code block with vertical strips, where each strip has constant thickness greater than $r$, the range of the stabilizer generators. Counting from the left edge, color the odd-numbered strips black and the even-numbered strips white. We claim that at least one of these strips supports a nontrivial logical operator. To prove the claim by contradiction, suppose that no nontrivial logical operator is supported on any of the black strips. Thus each black strip is correctable, and because the strips are separated by distance greater than $r$, the union of all black strips is also correctable and therefore cleanable. Consider a nontrivial logical operator; after cleaning this logical operator on the black strips, we obtain a nontrival logical operator $x$ supported on the union of all white strips. Thus $x$ commutes with all stabilizer generators. Because the while strips are separated by distance greater than $r$,

no stabilizer generator has support on more than one strip. It follows that all stabilizer generators commute with the restriction of $x$ to a single strip; that is, $x$ is a product of strip operators, each supported on a vertical strip of constant width, and each of which is logical (commutes with the stabilizer). Because $x$ is a nontrivial logical operator, at least one of these strip operators must be both logical and nontrivial. This contradicts our assumption that no strip supports a nontrivial logical operator, hence proving, as desired, that there is a nontrivial logical operator supported on a single vertical strip with constant thickness.

Of course, we did not need to choose the strips to be vertical. The same argument applies to any way of dividing the code block into regions that can be colored white and black, such that distinct black strips are separated by distance greater than $r$ and distinct white strips are separated by distance greater than $r$. In particular, it follows that there is a nontrivial logical operator supported on a *horizontal* strip of constant thickness as well.

### *7.20.2 2D local stabilizer codes are not self correcting*

Results that specify the geometry of a region that supports a logical operator have implications for the physical robustness of quantum memories.

Consider a stabilizer code with geometrically local generators $\{S_a\}$, where the qubits reside at the sites of a $D$-dimensional hypercubic lattice with linear size $L$. The generating set $\{S_a\}$ might be overcomplete, but we assume that the number of local generators acting nontrivially on each qubit is a constant independent of $L$. The local Hamiltonian

$$H = -\sum_a \frac{1}{2}\left(S_a - I\right), \tag{7.253}$$

has a $2^k$-fold degenerate ground state with energy $E = 0$, where $k$ is the number of protected qubits — each ground state is a simultaneous eigenstate with eigenvalue one of all elements of $\{S_a\}$. If a quantum memory governed by this Hamiltonian is subjected to thermal noise, how well protected is the $2^k$-dimensional code space?

If $|\psi\rangle$ is a zero-energy eigenstate of $H$ and $x$ is a Pauli operator, then $x|\psi\rangle$ is an eigenstate of $H$ with eigenvalue $E(x)$, where $E(x)$ is the number of elements of $\{S_a\}$ that anticommute with $x$. Thermal fluctuations may excite the memory, but excitations with energy cost $E$ are suppressed by the Boltzmann factor $e^{-E/\tau}$ where $\tau$ is the temperature (and Boltzmann's constant $k_B$ has been set to one). We suppose that the environment applies a sequence of weight-one Pauli operators to the system, so that the error history after $t$ steps can be described as a walk on the Pauli group, starting at the identity:

$$\{x_i \in P, i = 0, 1, 2, 3, \ldots t\}, \tag{7.254}$$

where $x_0 = I$, and $x_{i+1}x_i^{-1}$ has weight one. Let $\mathcal{P}(z)$ denote the set of all such walks, with any number of steps, that start at $I$ and terminate at $z \in P$. We define

$$\Delta(z) \equiv \min_{\gamma \in \mathcal{P}(z)} \max_{x \in \gamma} E(x), \tag{7.255}$$

the minimum energy barrier that must be surmounted by any walk that reaches Pauli operator $z$. Thus such walks occur with a probability per unit time suppressed by the

Boltzmann factor $e^{-\Delta(z)/\tau}$. We also define

$$\Delta_{\min} \equiv \min_{x \in S^{\perp} \setminus S} \Delta(x), \tag{7.256}$$

(7.257)

Here $\Delta_{\min}$ is the lowest energy barrier protecting any nontrivial logical operator (representing a nontrivial coset of $S^{\perp}/S$).

We say that a quantum memory is *self correcting* if $\Delta_{\min}$ grows faster than logarithmically with $L$. In that case *all* nontrivial logical operators are suppressed by a Boltzmann factor whose reciprocal grows super-polynomially with $L$. Though the Pauli walk may not be a particularly accurate description of noise in realistic systems, it allows us to define the notion of barrier height precisely, and to state the criteria for self correction simply. Furthermore, we expect the Boltzmann factor $e^{-\Delta_{\min}/\tau}$ suppressing the Pauli walk to provide a reasonable (though crude) estimate of the logical error rate for more realistic noise models, assuming that the system attains thermal equilibrium.

Our finding that 2D stabilizer codes always admit strip logical operators has implicatioins regarding the self correction properties of these codes. For any logical operator $x$ supported on a vertical strip of constant width, we may build a Pauli walk that starts at $I$ and ends at $x$ by starting at the bottom of the strip, and then building the vertical string operator row by row. At each stage of this walk, any "excited" local stabilizer generator $S_a$ such that $S_a = -1$ acts only on qubits which are within distance $r$ of the boundary of the walk, where $r$ is the range of the stabilizer generators The number of such qubits is $O(1)$ and the total number of stabilizer generators acting on these qubits is $O(1)$. Therefore, the energy cost of the partially completed walk, and hence $\Delta_{\min}$, are $O(1)$. We conclude the 2D stabilizer codes are not self correcting.

## 7.21 Reed–Muller codes

Another way to construct codes that can correct many errors is to invoke the CSS construction. Recall, in particular, the special case of that construction that applies to a classical code $C$ that is contained in its dual code (we then say that $C$ is "weakly self-dual"). In the CSS construction, there is a codeword associated with each coset of $C$ in $C^{\perp}$. Thus we obtain an $[[n, k, d]]$ quantum code, where $n$ is the length of $C$, $d$ is (at least) the distance of $C^{\perp}$, and $k = \dim C^{\perp} - \dim C$. Therefore, for the construction of CSS codes that correct many errors, we seek weakly self-dual classical codes with a large minimum distance.

One class of weakly self-dual classical codes are the Reed-Muller codes. Though these are not especially efficient, they are very convenient, because they are easy to encode, recovery is simple, and it is not difficult to explain their mathematical structure.[4]

To prepare for the construction of Reed-Muller codes, consider Boolean functions on $m$ bits,

$$f : \{0,1\}^m \to \{0,1\} \ . \tag{7.258}$$

There are $2^{2^m}$ such functions forming what we may regard as a binary vector space of dimension $2^m$. It will be useful to have a basis for this space. Recall (§6.1), that any Boolean function has a disjunctive normal form. Since the NOT of a bit $x$ is $1 - x$, and

[4] See, *e.g.*, MacWilliams and Sloane, Chapter 13.

the OR of two bits $x$ and $y$ can be expressed as

$$x \vee y == x + y - xy \ , \tag{7.259}$$

any of the Boolean functions can be expanded as a polynomial in the $m$ binary variables $x_{m-1}, x_{m-2}, \ldots, x_1, x_0$ . A basis for the vector space of polynomials consists of the $2^m$ functions

$$1, \ x_i, \ x_i x_j, \ x_i x_j x_k, \ \ldots, \tag{7.260}$$

(where, since $x^2 = x$, we may choose the factors of each monomial to be distinct). Each such function $f$ can be represented by a binary string of length $2^m$, whose value in the position labeled by the binary string $x_{m-1}x_{m-2}\ldots x_1x_0$ is $f(x_{m-1}, x_{m-2}, \ldots x_1, x_0)$. For example, for $m = 3$,

$$\begin{aligned} 1 &= (11111111) \\ x_0 &= (10101010) \\ x_1 &= (11001100) \\ x_2 &= (11110000) \\ x_0x_1 &= (10001000) \\ x_0x_2 &= (10100000) \\ x_1x_2 &= (11000000) \\ x_0x_1x_2 &= (10000000) \ . \end{aligned} \tag{7.261}$$

A subspace of this vector space is obtained if we restrict the degree of the polynomial to $r$ or less. This subspace is the Reed–Muller (or RM) code, denoted $R(r, m)$. Its length is $n = 2^m$ and its dimension is

$$k = 1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{r}. \tag{7.262}$$

Some special cases of interest are:

- $R(0, m)$ is the length-$2^m$ repetition code.
- $R(m-1, m)$ is the dual of the repetition code, the space on all length-$2^m$ even-weight strings.
- $R(1, 3)$ is the $n = 8$, $k = 4$ code spanned by $1, x_0, x_1, x_2$; it is in fact the $[8, 4, 4]$ extended Hamming code that we have already discussed.
- More generally, $R(m-2, m)$ is a $d = 4$ extended Hamming code for each $m \geq 3$. If we puncture this code (remove the last bit from all codewords) we obtain the $[n = 2^m - 1, k = n - m, d = 3]$ perfect Hamming code.
- $R(1, m)$ has $d = 2^{m-1} = \frac{1}{2}n$ and $k = m$. It is the dual of the extended Hamming code, and is known as a "first-order" Reed–Muller code. It is of considerable practical interest in its own right, both because of its large distance and because it is especially easy to decode.

We can compute the distance of the code $R(r, m)$ by invoking induction on $m$. First we must determine how $R(m+1, r)$ is related to $R(m, r)$. A function of $x_m, \ldots, x_0$ can be expressed as

$$f(x_m, \ldots, x_0) = g(x_{m-1}, \ldots, x_0) + x_m h(x_{m-1}, \ldots, x_0) \ , \tag{7.263}$$

and if $f$ has degree $r$, then $g$ must be of degree $r$ and $h$ of degree $r-1$. Regarding $f$ as a vector of length $2^{m+1}$, we have

$$f = (g|g) + (h|0) \tag{7.264}$$

where $g, h$ are vectors of length $2^m$. Consider the distance between $f$ and

$$f' = (g'|g') + (h'|0) \ . \tag{7.265}$$

For $h = h'$ and $f \neq f'$ this distance is $\mathrm{wt}(f - f') = 2 \cdot \mathrm{wt}(g - g') \geq 2 \cdot \mathrm{dist}\ (R(r, m))$; for $h \neq h'$ it is at least $\mathrm{wt}(h - h') \geq \ \mathrm{dist}\ (R(r-1, m))$. If $d(r, m)$ denotes the distance of $R(r, m)$, then we see that

$$d(r, m+1) = \min\left(2\ d(r, m), d(r-1, m)\right) \ . \tag{7.266}$$

Now we can show that $d(r, m) = 2^{m-r}$ by induction on m. To start with, we check that $d(r, m = 1) = 2^{1-r}$ for $r = 0, 1$; $R(1, 1)$ is the space of all length 2 strings, and $R(0, 1)$ is the length-2 repetition code. Next suppose that $d = 2^{m-r}$ for all $m \leq M$ and $0 \leq r \leq m$. Then we infer that

$$d(r, m+1) = \min(2^{m-r+1}, 2^{m-r+1}) = 2^{m-r+1}, \tag{7.267}$$

for each $1 \leq r \leq m$. It is also clear that $d(m+1, m+1) = 1$, since $R(m+1, m+1)$ is the space of all binary strings of length $2^{m+1}$, and that $d(0, m+1) = 2^{m+1}$, since $R(0, m+1)$ is the length-$2^{m+1}$ repetition code. This completes the inductive step, and proves $d(r, m) = 2^{m-r}$.

It follows, in particular, that $R(m-1, m)$ has distance 2, and therefore that the dual of $R(r, m)$ is $R(m-r-1, m)$. First we notice that the binomial coefficients $\binom{m}{j}$ sum to $2^m$, so that $R(m-r-1)$ has the right dimension to be $R(r, m)^\perp$. It suffices, then, to show that $R(m-r-1)$ is contained in $R(r, m)$. But if $f \in R(r, m)$ and $g \in R(m-r-1, m)$, their product is a polynomial of degree at most $m-1$, and is therefore in $R(m-1, m)$. Each vector in $R(m-1, m)$ has even weight, so the inner product $f \cdot g$ vanishes; hence $g$ is in the dual $R(v, m)^\perp$. This shows that

$$R(r, m)^\perp = R(m-r-1, m). \tag{7.268}$$

It is because of this nice duality property that Reed–Muller codes are well-suited for the CSS construction of quantum codes.

In particular, the Reed–Muller code is weakly self-dual for $r \leq m-r-1$, or $2r \geq, m-1$, and self-dual for $2r = m-1$. In the self-dual case, the distance is

$$d = 2^{m-r} = 2^{\frac{1}{2}(m+1)} = \sqrt{2n} \ , \tag{7.269}$$

and the number of encoded bits is

$$k = \frac{1}{2} n = 2^{m-1} \ . \tag{7.270}$$

These self-dual codes, for $m = 3, 5, 7$, have parameters

$$[8, 4, 4], \quad [32, 16, 8], \quad [128, 64, 16] \ . \tag{7.271}$$

(The $[8, 4, 4]$ code is the extended Hamming code as we have already noted.) Associated with these self-dual codes are the $k = 0$ quantum codes with parameters

$$[[8, 0, 4]], \quad [[32, 0, 8]], \quad [[128, 0, 16]] \ , \tag{7.272}$$

and so forth.

One way to obtain a $k = 1$ quantum code is to *puncture* the self-dual Reed–Muller code, that is, to delete one of the $n = 2^m$ bits from the code. (It turns out not to matter *which* bit we delete.) The classical punctured code has parameters $n = 2^m - 1$, $d = 2^{\frac{1}{2}(m-1)} - 1 = \sqrt{2(n+1)} - 1$, and $k = \frac{1}{2}(n+1)$. Furthermore, the dual of the punctured code is its even subcode. (The even subcode consists of those RM codewords for which the bit removed by the puncture is zero, and it follows from the self-duality of the RM code that these are orthogonal to all the words (both odd and even weight) of the punctured code.) From these punctured codes, we obtain, via the CSS construction, $k = 1$ quantum codes with parameters

$$[[7, 1, 3]], \quad [[31, 1, 7]], \quad [[127, 1, 15]] \ , \tag{7.273}$$

and so forth. The $[7, 4, 3]$ Hamming code is obtained by puncturing the $[8, 4, 4]$ RM code, and the corresponding $[7, 1, 3]$ QECC is of course Steane's code. These QECC's have a distance that increases like the square root of their length.

These $k = 1$ codes are not among the most efficient of the known QECC's. Nevertheless they are of special interest, since their properties are especially conducive to implementing fault-tolerant quantum gates on the encoded data, as we will see in Chapter 8. In particular, one useful property of the self-dual RM codes is that they are "doubly even" — all codewords have a weight that is an integral multiple of four.

Of course, we can also construct quantum codes with $k > 1$ by applying the CSS construction to the RM codes. For example $R(3, 6)$, with parameters

$$\begin{aligned} n &= 2^m = 64 \\ d &= 2^{m-r} = 8 \\ k &= 1 + 6 + \binom{6}{2} + \binom{6}{3} = 1 + 6 + 15 + 20 = 42 \ , \end{aligned} \tag{7.274}$$

is dual to $R(2, 6)$, with parameters

$$\begin{aligned} n &= 2^m = 64 \\ d &= 2^{m-r} = 16 \\ k &= 1 + 6 + \binom{6}{2} = 1 + 6 + 15 = 22 \ , \end{aligned} \tag{7.275}$$

and so the CSS construction yields a QECC with parameters

$$[[64, 20, 8]] \ . \tag{7.276}$$

Many other weakly self-dual codes are known and can likewise be employed.

## 7.22 The Golay Code

From the perspective of pure mathematics, the most important error-correcting code (classical or quantum) ever discovered is also one of the first ever described in a published article — the Golay code. Here we will briefly describe the Golay code, as it too can be transformed into a nice QECC via the CSS construction. (Perhaps this QECC is not really important enough to deserve a section of this chapter; still, I have included it just for fun.)

The (extended) Golay code is a self-dual $[24, 12, 8]$ classical code. If we puncture it

(remove any one of its 24 bits), we obtain the $[23, 12, 7]$ Golay code, which can correct three errors. This code is actually perfect, as it saturates the sphere-packing bound:

$$1 + \binom{23}{1} + \binom{23}{2} + \binom{23}{3} = 2^{11} = 2^{23-12}. \tag{7.277}$$

In fact, perfect codes that correct more than one error are extremely rare. It can be shown[5] that the *only* perfect codes (linear or nonlinear) over *any* finite field that can correct more than one error are the $[23, 12, 7]$ code and one other binary code discovered by Golay, with parameters $[11, 6, 5]$.

The $[24, 12, 8]$ Golay code has a very intricate symmetry. The symmetry is characterized by its automorphism group — the group of permutations of the 24 bits that take codewords to codewords. This is the Mathieu group $M_{24}$, a sporadic simple group of order 244,823,040 that was discovered in the 19th century.

The $2^{12} = 4096$ codewords have the weight distribution (in an obvious notation)

$$0^1 8^{759} 12^{2576} 16^{759} 24^1 \ . \tag{7.278}$$

Note in particular that each weight is a multiple of 4 (the code is doubly even). What is the significance of the number 759 (= 3.11.23)? In fact it is

$$\binom{24}{5} \Big/ \binom{8}{5} = 759, \tag{7.279}$$

and it arises for this combination reason: with each weight-8 codeword we associate the eight-element set ("octad") where the codeword has its support. Each 5-element subset of the 24 bits is contained in exactly one octad (a reflection of the code's large symmetry).

What makes the Golay code important in mathematics? Its discovery in 1949 set in motion a sequence of events that led, by around 1980, to a complete classification of the finite simple groups. This classification is one of the greatest achievements of 20th century mathematics.

(A group is simple if it contains no nontrivial normal subgroup. The finite simple groups may be regarded as the building blocks of all finite groups in the sense that for any finite group $G$ there is a unique decomposition of the form

$$G \equiv G_0 \supseteq G_1 \supseteq G_2 \geq \ldots \supseteq G_n, \tag{7.280}$$

where each $G_{j+1}$ is a normal subgroup of $G_j$, and each quotient group $G_j/G_{j+1}$ is simple. The finite simple groups can be classified into various infinite families, plus 26 additional "sporadic" simple groups that resist classification.)

The Golay code led Leech, in 1964, to discover an extraordinarily close packing of spheres in 24 dimensions, known as the *Leech Lattice* $\Lambda$. The lattice points (the centers of the spheres) are 24-component integer-valued vectors with these properties: to determine if $\vec{x} = (x_1, x_2 \ldots, x_{24})$ is contained in $\Lambda$, write each component $x_j$ in binary notation,

$$x_j = \ldots x_{j3} x_{j2} x_{j1} x_{j0} \ . \tag{7.281}$$

Then $\vec{x} \in \Lambda$ if

**(i)** The $x_{j0}$'s are either all 0's or all 1's.

[5] MacWilliams and Sloane §6.10.

**(ii)** The $x_{j2}$'s are an even parity 24-bit string if the $x_{j0}$'s are 0, and an odd parity 24-bit string if the $x_{j0}$'s are 1.

**(iii)** The $x_{j1}$'s are a 24-bit string contained in the Golay code.

When these rules are applied, a negative number is represented by its binary complement, *e.g.*

$$\begin{aligned} -1 &= \ldots 11111 \ , \\ -2 &= \ldots 11110 \ , \\ -3 &= \ldots 11101 \ , \\ &\text{etc.} \end{aligned} \tag{7.282}$$

We can easily check that $\Lambda$ is a lattice; that is, it is closed under addition. (Bits other than the last three in the binary expansion of the $x_j$'s are unrestricted).

We can now count the number of nearest neighbors to the origin (or the number of spheres that touch any given sphere). These points are all $(\text{distance})^2 = 32$ away from the origin:

$$\begin{aligned} (\pm 2)^8 &: 2^7 \cdot 759 \\ (\pm 3)(\mp 1)^{23} &: 2^{12} \cdot 24 \\ (\pm 4)^2 &: 2^2 \cdot \binom{24}{2} \ . \end{aligned} \tag{7.283}$$

That is, there are $759 \cdot 2^7$ neighbors that have eight components with the values $\pm 2$ — their support is on one of the 759 weight-8 Golay codewords, and the number of $-$ signs must be even. There are $2^{12} \cdot 24$ neighbors that have one component with value $\pm 3$ (this component can be chosen in 24 ways) and the remaining 23 components have the value $(\mp 1)$. If, say, $+3$ is chosen, then the position of the $+3$, together with the position of the $-1$'s, can be any of the $2^{11}$ Golay codewords with value 1 at the position of the $+3$. There are $2^2 \cdot \binom{24}{2}$ neighbors with two components each taking the value $\pm 4$ (the signs are unrestricted). Altogether, the coordination number of the lattice is $196,560$.

The Leech lattice has an extraordinary automorphism group discovered by Conway in 1968. This is the finite subgroup of the 24-dimensional rotation group $SO(24)$ that preserves the lattice. The order of this finite group (known as $\cdot 0$, or "dot oh") is

$$2^{22} \cdot 3^9 \cdot 5^4 \cdot 7^2 \cdot 11 \cdot 13 \cdot 23 = 8,315,553,613,086,720,000 \simeq 8.3 \times 10^{18}. \tag{7.284}$$

If its two element center is modded out, the sporadic simple group $\cdot 1$ is obtained. At the time of its discovery, $\cdot 1$ was the largest of the sporadic simple groups that had been constructed.

The Leech lattice and its automorphism group eventually (by a route that won't be explained here) led Griess in 1982 to the construction of the most amazing sporadic simple group of all (whose existence had been inferred earlier by Fischer and Griess). It is a finite subgroup of the rotation group in 196,883 dimensions, whose order is approximately $8.08 \times 10^{53}$. This behemoth known as $F_1$ has earned the nickname "the monster" (though Griess prefers to call it "the friendly giant".) It is the largest of the sporadic simple groups, and the last to be discovered.

Thus the classification of the finite simple groups owes much to (classical) coding theory, and to the Golay code in particular. Perhaps the theory of QECC's can also bequeath to mathematics something of value and broad interest!

Anyway, since the (extended) $[24, 12, 8]$ Golay code is self-dual, the $[23, 12, 7]$ code obtained by puncturing it is weakly self dual; its $[23, 11, 8]$ dual is its even subcode. From it, a $[23, 1, 7]$ QECC can be constructed by the CSS method. This code is not the most efficient quantum code that can correct three errors (there is a $[17, 1, 7]$ code that saturates the Rains bound), but it has especially nice properties that are conducive to fault-tolerant quantum computation, as we will see in Chapter 8.

## 7.23 The Quantum Channel Capacity

As we have formulated it up until now, our goal in constructing quantum error correcting codes has been to maximize the distance $d$ of the code, given its length $n$ and the number $k$ of encoded qubits. Larger distance provides better protection against errors, as a distance $d$ code can correct $d-1$ erasures, or $(d-1)/2$ errors at unknown locations. We have observed that "good" codes can be constructed, that maintain a finite rate $k/n$ for $n$ large, and correct a number of errors $pn$ that scales linearly with $n$.

Now we will address a related but rather different question about the asymptotic performance of QECC's. Consider a superoperator \$ that acts on density operators in a Hilbert space $\mathcal{H}$. Now consider \$ acting independently each copy of $\mathcal{H}$ contained in the $n$-fold tensor product

$$\mathcal{H}^{(n)} = \mathcal{H} \otimes \ldots \otimes \mathcal{H}. \tag{7.285}$$

We would like to select a code subspace $\mathcal{H}^{(n)}_{\rm code}$ of $\mathcal{H}^{(n)}$ such that quantum information residing in $\mathcal{H}^{(n)}_{\rm code}$ can be subjected to the superoperator

$$\$^{(n)} = \$ \otimes \ldots \otimes \$, \tag{7.286}$$

and yet can still be decoded with high fidelity.

The rate of a code is defined as

$$R = \frac{\log \mathcal{H}^{(n)}_{\rm code}}{\log \mathcal{H}^{(n)}} \ ; \tag{7.287}$$

this is the number of qubits employed to carry one qubit of encoded information. The *quantum channel capacity* $Q(\$)$ of the superoperator \$ is the maximum asymptotic rate at which quantum information can be sent over the channel with arbitrarily good fidelity. That is, $Q(\$)$ is the largest number such that for any $R < Q(\$)$ and any $\varepsilon > 0$, there is a code $\mathcal{H}^{(n)}_{\rm code}$ with rate at least $R$, such that for any $|\psi\rangle \in \mathcal{H}^{(n)}_{\rm code}$, the state $\boldsymbol{\rho}$ recovered after $|\psi\rangle$ passes through $\$^{(n)}$ has fidelity

$$F = \langle\psi|\boldsymbol{\rho}|\psi\rangle > 1 - \varepsilon. \tag{7.288}$$

Thus, $Q(\$)$ is a quantum version of the capacity defined by Shannon for a classical noisy channel. As we have already seen in Chapter 5, this $Q(\$)$ is not the only sort of capacity that can be associated with a quantum channel. It is also of considerable interest to ask about $C(\$)$, the maximum rate at which *classical* information can be transmitted through a quantum channel with arbitrarily small probability of error. A formal answer to this question was formulated in §5.4, but only for a restricted class of possible encoding schemes; the general answer is still unknown. The quantum channel capacity $Q(\$)$ is even less well understood than the classical capacity $C(\$)$ of a quantum channel. Note that $Q(\$)$ is not the same thing as the maximum asymptotic rate $k/n$

that can be achieved by "good" $[[n, k, d]]$ QECC's with positive $d/n$. In the case of the quantum channel capacity we need not insist that the code correct *any* possible distribution of $pn$ errors, as long as the errors that cannot be corrected become highly atypical for $n$ large.

Here we will mostly limit the discussion to two interesting examples of quantum channels acting on a single qubit — the quantum erasure channel (for which $Q$ is exactly known), and the depolarizing channel (for which $Q$ is still unknown, but useful upper and lower bounds can be derived).

What are these channels? In the case of the quantum erasure channel, a qubit transmitted through the channel either arrives intact, or (with probability $p$) becomes lost and is never received. We can find a unitary representation of this channel by embedding the qubit in the three-dimensional Hilbert space of a qubit with orthonormal basis $\{|0\rangle, |1\rangle, |2\rangle\}$. The channel acts according to

$$\begin{aligned} |0\rangle \otimes |0\rangle_E &\to \sqrt{1-p}|0\rangle \otimes |0\rangle_E + \sqrt{p}|2\rangle \otimes |1\rangle_E, \\ |1\rangle \otimes |0\rangle_E &\to \sqrt{1-p}|1\rangle \otimes |0\rangle_E + \sqrt{p}|2\rangle \otimes |2\rangle_E, \end{aligned} \tag{7.289}$$

where $\{|0\rangle_E, |1\rangle_E, |2\rangle_E\}$ are mutually orthogonal states of the environment. The receiver can measure the observable $|2\rangle\langle 2|$ to determined whether the qubit is undamaged or has been "erased."

The depolarizing channel (with error probability $p$) was discussed at length in §3.4.1. We see that, for $p \leq 3/4$, we may describe the fate of a qubit transmitted through the channel this way: with probability $1-q$ (where $q = 4/3p$), the qubit arrives undamaged, and with probability $q$ it is *destroyed*, in which case it is described by the random density matrix $\frac{1}{2}\mathbf{1}$.

Both the erasure channel and the depolarizing channel destroy a qubit with a specified probability. The crucial difference between the two channels is that in the case of the erasure channel, the receiver knows which qubits have been destroyed; in the case of the depolarizing channel, the damaged qubits carry no identifying marks, which makes recovery more challenging. Of course, for both channels, the sender has no way to know ahead of time which qubits will be obliterated.

### *7.23.1 Erasure channel*

The quantum channel capacity of the erasure channel can be precisely determined. First we will derive an upper bound on $Q$, and then we will show that codes exist that achieve high fidelity and attain a rate arbitrarily close to the upper bound.

As the first step in the derivation of an upper bound on the capacity, we show that $Q = 0$ for $p > \frac{1}{2}$.

– Figure –

We observe that the erasure channel can be realized if Alice sends a qubit to Bob, and a third party Charlie decides at random to either *steal* the qubit (with probability $p$) or allow the qubit to pass unscathed to Bob (with probability $1-p$).

If Alice sends a large number $n$ of qubits, then about $(1-p)n$ reach Bob, and $pn$

are intercepted by Charlie. Hence for $p > \frac{1}{2}$, Charlie winds up in possession of more qubits than Bob, and if Bob can recover the quantum information encoded by Alice, then certainly Charlie can as well. Therefore, if $Q(p) > 0$ for $p > \frac{1}{2}$, Bob and Charlie can clone the unknown encoded quantum states sent by Alice, which is impossible. (Strictly speaking, they can clone with fidelity $F = 1 - \varepsilon$, for any $\varepsilon > 0$.) We conclude that $Q(p) = 0$ for $p > \frac{1}{2}$.

To obtain a bound on $Q(p)$ in the case $p < \frac{1}{2}$, we will appeal to the following lemma. Suppose that Alice and Bob are connected by both a perfect noiseless channel and a noisy channel with capacity $Q > 0$. And suppose that Alice sends $m$ qubits over the perfect channel and $n$ qubits over the noisy channel. Then the number $r$ of encoded qubits that Bob may recover with arbitrarily high fidelity must satisfy

$$r \leq m + Qn. \tag{7.290}$$

We derive this inequality by noting that Alice and Bob can simulate the $m$ qubits sent over the perfect channel by sending $m/Q$ over the noisy channel and so achieve a rate

$$R = \frac{r}{m/Q + n} = \left(\frac{r}{m + Qn}\right) Q, \tag{7.291}$$

over the noisy channel. Were $r$ to exceed $m + Qn$, this rate $R$ would exceed the capacity, a contradiction. Therefore eq. (7.290) is satisfied.

How consider the erasure channel with error probability $p_1$, and suppose $Q(p_1) > 0$. Then we can bound $Q(p_2)$ for $p_2 \leq p_1$ by

$$Q(p_2) \leq 1 - \frac{p_2}{p_1} + \frac{p_2}{p_1} Q(p_1). \tag{7.292}$$

(In other words, if we plot $Q(p)$ in the $(p, Q)$ plane, and we draw a straight line segment from any point $(p_1, Q_1)$ on the plot to the point $(p = 0, Q = 1)$, then the curve $Q(p)$ must lie on or below the segment in the interval $0 \leq p \leq p_1$; if $Q(p)$ is twice differentiable, then its second derivative cannot be positive.) To obtain this bound, imagine that Alice sends $n$ qubits to Bob, knowing ahead of time that $n(1 - p_2/p_1)$ specified qubits will arrive safely. The remaining $n(p_2/p_1)$ qubits are erased with probability $p_1$. Therefore, Alice and Bob are using both a perfect channel and an erasure channel with erasure probability $p_1$; eq. (7.290) holds, and the rate $R$ they can attain is bounded by

$$R \leq 1 - \frac{p_2}{p_1} + \frac{p_2}{p_1} Q(p_1). \tag{7.293}$$

On the other hand, for $n$ large, altogether about $np_2$ qubits are erased, and $(1 - p_2)n$ arrive safely. Thus Alice and Bob have an erasure channel with erasure probability $p_2$, except that they have the additional advantage of knowing ahead of time that some of the qubits that Alice sends are invulnerable to erasure. With this information, they can be no worse off than without it; eq. (7.292) then follows. The same bound applies to the depolarizing channel as well.

Now, the result $Q(p) = 0$ for $p > 1/2$ can be combined with eq. (7.292). We conclude that the curve $Q(p)$ must be on or below the straight line connecting the points $(p = 0, Q = 1)$ and $(p = 1/2, Q = 0)$, or

$$Q(p) \leq 1 - 2p, \quad 0 \leq p \leq \frac{1}{2}. \tag{7.294}$$

In fact, there are stabilizer codes that actually attain the rate $1 - 2p$ for $0 \leq p \leq$

1/2. We can see this by borrowing an idea from Claude Shannon, and averaging over random stabilizer codes. Imagine choosing, in succession, altogether $n-k$ stabilizer generators. Each is selected from among the $4^n$ Pauli operators, where all have equal a priori probability, except that each generator is required to commute with all generators chosen in previous rounds.

Now Alice uses this stabilizer code to encode an arbitrary quantum state in the $2^k$-dimensional code subspace, and sends the $n$ qubits to Bob over an erasure channel with erasure probability $p$. Will Bob be able to recover the state sent by Alice?

Bob replaces each erased qubit by a qubit in the state $|0\rangle$, and then proceeds to measure all $n-k$ stabilizer generators. From this syndrome measurement, he hopes to infer the Pauli operator $\boldsymbol{E}$ acting on the replaced qubits. Once $\boldsymbol{E}$ is known, we can apply $\boldsymbol{E}^\dagger$ to recover a perfect duplicate of the state sent by Alice. For $n$ large, the number of qubits that Bob must replace is about $pn$, and he will recover successfully if there is a unique Pauli operator $\boldsymbol{E}$ that can produce the syndrome that he finds. If more than one Pauli operator acting on the replaced qubits has this same syndrome, then recovery may fail.

How likely is failure? Since there are about $pn$ replaced qubits, there are about $4^{pn}$ Pauli operators with support on these qubits. Furthermore, for any particular Pauli operator $\boldsymbol{E}$, a random stabilizer code generates a random syndrome — each stabilizer generator has probability 1/2 of commuting with $\boldsymbol{E}$, and probability 1/2 of anti-commuting with $\boldsymbol{E}$. Therefore, the probability that two Pauli operators have the same syndrome is $(1/2)^{n-k}$.

There is at least one particular Pauli operator acting on the replaced qubits that has the syndrome found by Bob. But the probability that another Pauli operator has this same syndrome (and hence the probability of a recovery failure) is no worse than

$$P_{\text{fail}} \leq 4^{pn} \left(\frac{1}{2}\right)^{n-k} = 2^{-n(1-2p-R)}. \tag{7.295}$$

where $R = k/n$ is the rate. Eq. (7.295) bounds the failure probability if we *average* over all stabilizer codes with rate $R$; it follows that at least one particular stabilizer code must exist whose failure probability also satisfies the bound.

For that particular code, $P_{\text{fail}}$ gets arbitrarily small as $n \to \infty$, for any rate $R = 1-2p-\delta$ strictly less than $1-2p$. Therefore $R = 1-2p$ is asymptotically attainable; combining this result with the inequality eq. (7.294) we obtain the capacity of the quantum erasure channel:

$$Q(p) = 1-2p, \quad 0 \leq p \leq \frac{1}{2}\ . \tag{7.296}$$

If we wanted assurance that a distinct syndrome could be assigned to all ways of damaging $pn$ erased qubits, then we would require an $[[n,k,d]]$ quantum code with distance $d > pn$. Our Gilbert–Varshamov bound of §7.16 guarantees the existence of such a code for

$$R < 1 - H_2(p) - p\log_2 3. \tag{7.297}$$

This rate can be achieved by a code that recovers from any of the possible ways of erasing up to $pn$ qubits. It lies strictly below the capacity for $p > 0$, because to achieve high average fidelity, it suffices to be able to correct the *typical* erasures, rather than all possible erasures.

### 7.23.2 Depolarizing channel

The capacity of the depolarizing channel is still not precisely known, but we can obtain some interesting upper and lower bounds.

As for the erasure channel, we can find an upper bound on the capacity by invoking the no-cloning theorem. Recall that for the depolarizing channel with error probability $p < 3/4$, each qubit either passes safely with probability $1-4/3p$, or is randomized (replaced by the maximally mixed state $\boldsymbol{\rho} = \frac{1}{2}\mathbf{1}$) with probability $q = 4/3p$. An eavesdropper Charlie, then, can simulate the channel by intercepting qubits with probability $q$, and replacing each stolen qubit with a maximally mixed qubit. For $q > 1/2$, Charlie steals more than half the qubits and is in a better position than Bob to decode the state sent by Alice. Therefore, to disallow cloning, the rate at which quantum information is sent from Alice to Bob must be strictly zero for $q > 1/2$ or $p > 3/8$:

$$Q(p) = 0, \quad p > \frac{3}{8}. \tag{7.298}$$

In fact we can obtain a stronger bound by noting that Charlie can choose a better eavesdropping strategy – he can employ the optimal *approximate* cloner that you studied in a homework problem. This device, applied to each qubit sent by Alice, replaces it by two qubits that each approximate the original with fidelity $F = 5/6$, or

$$|\psi\rangle\langle\psi| \to \left[(1-q)|\psi\rangle\langle\psi| + q\frac{1}{2}\mathbf{1}\right]^{\otimes 2}, \tag{7.299}$$

where $F = 5/6 = 1 - 1/2q$. By operating the cloner, both Charlie and Bob can receive Alice's state transmitted through the $q = 1/3$ depolarizing channel. Therefore, the attainable rate must vanish; otherwise, by combining the approximate cloner with quantum error correction, Bob and Charlie would be able to clone Alice's unknown state *exactly*. We conclude that the capacity vanishes for $q > 1/3$ or $p > 1/4$:

$$Q(p) = 0, \quad p > \frac{1}{4}. \tag{7.300}$$

Invoking the bound eq. (7.292) we infer that

$$Q(p) \leq 1 - 4p, \quad 0 \leq p \leq \frac{1}{4}. \tag{7.301}$$

This result actually coincides with our bound on the rate of $[[n, k, d]]$ codes with $k \geq 1$ and $d \geq 2pn + 1$ found in §7.9. A bound on the capacity is *not* the same thing as a bound on the allowable error probability for an $[[n, k, d]]$ code (and in the latter case the Rains bound is tighter). Still, the similarity of the two results bound may not be a complete surprise, as both bounds are derived from the no-cloning theorem.

We can obtain a lower bound on the capacity by estimating the rate that can be attained through random stabilizer coding, as we did for the erasure channel. Now, when Bob measures the $n - k$ (randomly chosen, commuting) stabilizer generators, he hopes to obtain a syndrome that points to a unique one among the typical Pauli error operators that can arise with nonnegligible probability when the depolarizing channel acts on the $n$ qubits sent by Alice. The number $N_{\text{typ}}$ of typical Pauli operators with total probability $1 - \varepsilon$ can be bounded by

$$N_{\text{typ}} \leq 2^{n(H_2(p)+p\log_2 3+\delta)}, \tag{7.302}$$

for any $\delta, \varepsilon > 0$ and $n$ sufficiently large. Bob's attempt at recovery can fail if another

among these typical Pauli operators has the same syndrome as the actual error operator. Since a random code assigns a random $(n-k)$-bit syndrome to each Pauli operator, the failure probability can be bounded as

$$P_{\text{fail}} \leq 2^{n(H_2(p)+p\log_2 3+\delta)} 2^{k-n} + \varepsilon \ . \tag{7.303}$$

Here the second term bounds the probability of an atypical error, and the first bounds the probability of an ambiguous syndrome in the case of a typical error. We see that the failure probability, averaged over random stabilizer codes, becomes arbitrarily small for large $n$, for any $\delta' < 0$ and rate $R$ such that

$$R \equiv \frac{k}{n} \ < \ 1 - H_2(p) - p\log_2 3 - \delta'. \tag{7.304}$$

If the failure probability, averaged over codes, is small, there is a particular code with small failure probability, and we conclude that the rate $R$ is attainable; the capacity of the depolarizing channel is bounded below as

$$Q(p) \ \geq \ 1 - H_2(p) - p\log_2 3 \ . \tag{7.305}$$

Not coincidentally, the rate attainable by random coding agrees with the asymptotic form of the quantum Hamming upper bound on the rate of nondegenerate $[[n,k,d]]$ codes with $d > 2pn$; we arrive at both results by assigning a distinct syndrome to each of the typical errors. Of course, the Gilbert–Varshamov lower bound on the rate of $[[n,k,d]]$ codes lies below $Q(p)$, as it is obtained by demanding that the code can correct *all* the errors of weight $pn$ or less, not just the typical ones.

This random coding argument can also be applied to a somewhat more general channel, in which $\boldsymbol{X}, \boldsymbol{Y}$, and $\boldsymbol{Z}$ errors occur at different rates. (We'll call this a "Pauli channel.") If an $\boldsymbol{X}$ error occurs with probability $p_X$, a $\boldsymbol{Y}$ error with probability $p_Y$, a $\boldsymbol{Z}$ error with probability $p_Z$, and no error with probability $p_I \equiv 1 - p_X - p_Y - p_Z$, then the number of typical errors on $n$ qubits is

$$\frac{n!}{(p_X n)!(p_Y n)!(p_Z n)!(p_I n)!} \sim 2^{nH(p_I,p_X,p_Y,p_Z)}, \tag{7.306}$$

where

$$H \equiv H(p_I,p_X,p_Y,p_Z) = -p_I\log_2 p_I - p_X\log_2 p_X - p_Y\log_2 p_Y - p_Z\log_2 p_Z, \tag{7.307}$$

is the Shannon entropy of the probability distribution $\{p_I, p_X, p_Y, p_Z\}$. Now we find

$$Q(p_I,p_X,p_Y,p_Z) \ \geq \ 1 - H(p_I,p_X,p_Y,p_Z) \ ; \tag{7.308}$$

if the rate $R$ satisfies $R < 1 - H$, then again it is highly unlikely that a single syndrome of a random stabilizer code will point to more than one typical error operator.

### *7.23.3 Degeneracy and capacity*

Our derivation of a lower bound on the capacity of the depolarizing channel closely resembles the argument in §5.1.3 for a lower bound on the capacity of the classical binary symmetric channel. In the classical case, there was a matching upper bound. If the rate were larger, then there would not be enough syndromes to attach to all of the typical errors.

In the quantum case, the derivation of the matching upper bound does not carry

through, because a quantum code can be degenerate. We may not need a distinct syndrome for each typical error, as some of the possible errors could act trivially on the code subspace. Indeed, not only does the derivation fail; the matching upper bound is actually false – rates exceeding $1 - H_2(p) - p \log_2 3$ actually *can* be attained.[6]

Shor and Smolin investigated the rate that can be achieved by concatenated codes, where the outer code is a random stabilizer code, and the inner code is a degenerate code with a relatively small block size. Their idea is that the degeneracy of the inner code will allow enough typical errors to act trivially in the code space that a higher rate can be attained than through random coding alone.

To investigate this scheme, imagine that encoding and decoding are each performed in two stages. In the first stage, using the (random) outer code that she and Bob have agreed on, Alice encodes the state that she has selected in a large $n$-qubit block. In the second stage, Alice encodes each of these $n$-qubits in a block of $m$ qubits, using the inner code. Similarly, when Bob receives the $nm$ qubits, he first decodes each inner block of $m$, and then subsequently decodes the block of $n$.

We can evidently describe this procedure in an alternative language — Alice and Bob are using just the outer code, but the qubits are being transmitted through a composite channel.

– Figure –

This modified channel consists (as shown) of: first the inner encoder, then propagation through the original noisy channel, and finally inner decoding and inner recovery. The rate that can be attained through the original channel, via concatenated coding, is the same as the rate that can be attained through the modified channel, via random coding.

Specifically, suppose that the inner code is an $m$-qubit repetition code, with stabilizer

$$\boldsymbol{Z}_1\boldsymbol{Z}_2,\ \boldsymbol{Z}_1\boldsymbol{Z}_3,\ \boldsymbol{Z}_1\boldsymbol{Z}_4, \ldots, \boldsymbol{Z}_1\boldsymbol{Z}_m. \tag{7.309}$$

This is not much of a quantum code; it has distance 1, since it is insensitive to phase errors — each $\boldsymbol{Z}_j$ commutes with the stabilizer. But in the present context its important feature is it high degeneracy, all $\boldsymbol{Z}_i$ errors are equivalent.

The encoding (and decoding) circuit for the repetition code consists of just $m - 1$ CNOT's, so our composite channel looks like (in the case $m = 3$)

– Figure –

where \$ denotes the original noisy channel. (We have also suppressed the final recovery step of the decoding; *e.g.*, if the measured qubits both read 1, we should flip the data qubit. In fact, to simplify the analysis of the composite channel, we will dispense with this step.)

Since we recall that a CNOT propagates bit flips forward (from control to target) and

[6] P.W. Shor and J.A. Smolin, "Quantum Error-Correcting Codes Need Not Completely Reveal the Error Syndrome" quant-ph/9604006; D.P. DiVincen, P.W. Shor, and J.A. Smolin, "Quantum Channel Capacity of Very Noisy Channels," quant-ph/9706061.

phase flips backward (from target to control), we see that for each possible measurement outcome of the auxiliary qubits, the composite channel is a Pauli channel. If we imagine that this measurement of the $m-1$ inner block qubits is performed for each of the $n$ qubits of the outer block, then Pauli channels act independently on each of the $n$ qubits, but the channels acting on different qubits have different parameters (error probabilities $p_I^{(i)}, p_X^{(i)}, p_Y^{(i)}, p_Z^{(i)}$ for the $i$th qubit). Now the number of typical error operators acting on the $n$ qubits is

$$2^{\sum_{i=1}^n H_i} \tag{7.310}$$

where

$$H_i = H(p_I^{(i)}, p_X^{(i)}, p_Y^{(i)}, p_Z^{(i)}), \tag{7.311}$$

is the Shannon entropy of the Pauli channel acting on the $i$th qubit. By the law of large numbers, we will have

$$\sum_{i=1}^n H_i = n\langle H\rangle, \tag{7.312}$$

for large $n$, where $\langle H\rangle$ is the Shannon entropy, averaged over the $2^{m-1}$ possible classical outcomes of the measurement of the extra qubits of the inner code. Therefore, the rate that can be attained by the random outer code is

$$R = \frac{1-\langle H\rangle}{m}, \tag{7.313}$$

(we divide by $m$, because the concatenated code has a length $m$ times longer than the random code).

Shor and Smolin discovered that there are repetition codes (values of $m$) for which, in a suitable range of $p$, $1-\langle H\rangle$ is positive while $1-H_2(p)-p\log_2 3$ is negative. In this range, then, the capacity $Q(p)$ is nonzero, showing that the lower bound eq. (7.305) is not tight.

A nonvanishing asymptotic rate is attainable through random coding for $1-H_2(p)-p\log_2 3>0$, or $p<p_{\max}\simeq .18929$. If a random outer code is concatenated with a 5-qubit inner repetition code ($m=5$ turns out to be the optimal choice), then $1-\langle H\rangle>0$ for $p<p'_{\max}\simeq .19036$; the maximum error probability for which a nonzero rate is attainable increases by about 0.6%. It is not obvious that the concatenated code should outperform the random code in this range of error probability, though as we have indicated, it might have been expected because of the (phase) degeneracy of the repetition code. Nor is it obvious that $m=5$ should be the best choice, but this can be verified by an explicit calculation of $\langle H\rangle$.[7]

The depolarizing channel is one of the very simplest of quantum channels. Yet even for this case, the problem of characterizing and calculating the capacity is largely unsolved. This example illustrates that, due to the possibility of degenerate coding, the capacity problem is considerably more subtle for quantum channels than for classical channels.

We have seen that (if the errors are well described by the depolarizing channel), quantum information can be recovered from a quantum memory with arbitrarily high fidelity, as long as the probability of error per qubit is less than 19%. This is an improvement relative to the 10% error rate that we found could be handled by concatenation of the

[7] In fact a very slight further improvement can be achieved by concatenating a random code with the 25-qubit generalized Shor code described in the exercises – then a nonzero rate is attainable for $p<p''_{\max}\simeq .19056$ (another 0.1% better than the maximum tolerable error probability with repetition coding).

$[[5,1,3]]$ code. In fact $[[n,k,d]]$ codes that can recover from any distribution of up to $pn$ errors do not exist for $p > 1/6$, according to the Rains bound. Nonzero capacity is possible for error rates between 16.7% and 19% because it is sufficient for the QECC to be able to correct the typical errors rather than all possible errors.

However, the claim that recovery is possible even if 19% of the qubits sustain damage is highly misleading in an important respect. This result applies if encoding, decoding, and recovery can be executed flawlessly. But these operations are actually very intricate quantum computations that in practice will certainly be susceptible to error. We will not fully understand how well coding can protect quantum information from harm until we have learned to design an error recovery protocol that is robust even if the execution of the protocol is flawed. Such *fault-tolerant* protocols will be developed in Chapter 8.

## 7.24 Summary

**Quantum error-correcting codes:** Quantum error correction can protect quantum information from both decoherence and "unitary errors" due to imperfect implementations of quantum gates. In a (binary) *quantum error-correcting code* (QECC), the $2^k$-dimensional Hilbert space $\mathcal{H}_{\text{code}}$ of $k$ encoded qubits is embedded in the $2^n$-dimensional Hilbert space of $n$ qubits. Errors acting on the $n$ qubits are reversible provided that $\langle\psi|\boldsymbol{M}_\nu^\dagger \boldsymbol{M}_\mu|\psi\rangle/\langle\psi|\psi\rangle$ is independent of $|\psi\rangle$ for any $|\psi\rangle \in \mathcal{H}_{\text{code}}$ and any two Kraus operators $\boldsymbol{M}_{\mu,\nu}$ occuring in the expansion of the error superoperator. The recovery superoperator transforms entanglement of the environment with the code block into entanglement of the environment with an ancilla that can then be discarded.

**Quantum stabilizer codes:** Most QECC's that have been constructed are *stabilizer codes*. A binary stabilizer code is characterized by its stabilizer $S$, an abelian subgroup of the $n$-qubit *Pauli group* $G_n = \{\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}^{\otimes n}$ (where $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ are the single-qubit Pauli operators). The code subspace is the simultaneous eigenspace with eigenvalue one of all elements of $S$; if $S$ has $n-k$ independent generators, then there are $k$ encoded qubits. A stabilizer code can correct each error in a subset $\mathcal{E}$ of $G_n$ if for each $\boldsymbol{E}_a, \boldsymbol{E}_b \in \mathcal{E}$, $\boldsymbol{E}_a^\dagger \boldsymbol{E}_b$ either lies in the stabilizer $S$ or outside of the normalizer $S^\perp$ of the stabilizer. If some $\boldsymbol{E}_a^\dagger \boldsymbol{E}_b$ is in $S$ for $\boldsymbol{E}_{a,b} \in \mathcal{E}$ the code is *degenerate*; otherwise it is *nondegenerate*. Operators in $S^\perp \setminus S$ are "logical" operators that act on encoded quantum information. The stabilizer $S$ can be associated with an additive code over the finite field $GF(4)$ that is self-orthogonal with respect to a symplectic inner product. The *weight* of a Pauli operator is the number of qubits on which its action is nontrivial, and the distance $d$ of a stabilizer code is the minimum weight of an element of $S^\perp \setminus S$. A code with length $n$, $k$ encoded qubits, and distance $d$ is called an $[[n,k,d]]$ quantum code. If the code enables recovery from any error superoperator with support on Pauli operators of weight $t$ or less, we say that the code "can correct $t$ errors." A code with distance $d$ can correct $[(d-1)/2]$ in unknown locations or $d-1$ errors in known locations. "Good" families of stabilizer codes can be constructed in which $d/n$ and $k/n$ remain bounded away from zero as $n \to \infty$.

**Examples:** The code of minimal length that can correct one error is a $[[5,1,3,]]$ quantum code associated with a classical $GF(4)$ Hamming code. Given a classical linear code $C_1$ and subcode $C_2 \subseteq C_1$, a Calderbank-Shor-Steane (CSS) quantum code can be constructed with $k = \dim(C_1) - \dim(C_2)$ encoded qubits. The distance $d$ of the CSS code satisfies $d \geq \min(d_1, d_2^\perp)$, where $d_1$ is the distance of $C_1$ and $d_2^\perp$ is the distance

of $C_2^\perp$, the dual of $C_2$. The simplest CSS code is a $[[7,1,3]]$ quantum code constructed from the $[7,4,3]$ classical Hamming code and its even subcode. An $[[n_1,1,d_1]]$ quantum code can be *concatenated* with an $[[n_2,1,d_2]]$ code to obtain a degenerate $[[n_1 n_2,1,d]]$ code with $d \geq d_1 d_2$.

**Quantum channel capacity:** The quantum channel capacity of a superoperator (noisy quantum channel) is the maximum rate at which quantum information can be transmitted over the channel and decoded with arbitrarily good fidelity. The capacity of the binary quantum erasure channel with erasure probability $p$ is $Q(p) = 1 - 2p$, for $0 \leq p \leq 1/2$. The capacity of the binary depolarizing channel is no yet known. The problem of calculating the capacity is subtle because the optimal code may be degenerate; in particular, random codes do not attain an asymptotically optimal rate over a quantum channel.

## Exercises

### 7.1 Phase error-correcting code

*a*) Construct stabilizer generators for an $n = 3$, $k = 1$ code that can correct a single bit flip; that is, ensure that recovery is possible for any of the errors in the set $\mathcal{E} = \{\boldsymbol{III}, \boldsymbol{XII}, \boldsymbol{IXI}, \boldsymbol{IIX}\}$. Find an orthonormal basis for the two-dimensional code subspace.

*b*) Construct stabilizer generators for an $n = 3$, $k = 1$ code that can correct a single phase error; that is, ensure that recovery is possible for any of the errors in the set $\mathcal{E} = \{\boldsymbol{III}, \boldsymbol{ZII}, \boldsymbol{IZI}, \boldsymbol{IIZ}\}$. Find an orthonormal basis for the two-dimensional code subspace.

### 7.2 Error-detecting codes

*a*) Construct stabilizer generators for an $[[n,k,d]] = [[3,0,2]]$ quantum code. With this code, we can detect any single-qubit error. Find the encoded state. (Does it look familiar?)

*b*) Two QECC's $C_1$ and $C_2$ (with the same length $n$) are *equivalent* if a permutation of qubits, combined with single-qubit unitary transformations, transforms the code subspace of $C_1$ to that of $C_2$. Are all $[[3,0,2]]$ stabilizer codes equivalent?

*c*) Does a $[[3,1,2]]$ stabilizer code exist?

### 7.3 Maximal entanglement

Consider the $[[5,1,3]]$ quantum code, whose stabilizer generators are $M_1 = \boldsymbol{XZZXI}$, and $M_{2,3,4}$ obtained by cyclic permutations of $M_1$, and choose the encoded operation $\bar{\boldsymbol{Z}}$ to be $\bar{\boldsymbol{Z}} = \boldsymbol{ZZZZZ}$. From the encoded states $|\bar{0}\rangle$ with $\bar{\boldsymbol{Z}}|\bar{0}\rangle = |\bar{0}\rangle$ and $|\bar{1}\rangle$ with $\bar{\boldsymbol{Z}}|\bar{1}\rangle = -|\bar{1}\rangle$, construct the $n = 6$, $k = 0$ code whose encoded state is

$$\frac{1}{\sqrt{2}}\left(|0\rangle \otimes |\bar{0}\rangle + |1\rangle \otimes |\bar{1}\rangle\right) \ . \tag{7.314}$$

*a*) Construct a set of stabilizer generators for this $n = 6$, $k = 0$ code.

*b*) Find the distance of this code. (Recall that for a $k = 0$ code, the distance is defined as the minimum weight of any element of the stabilizer.)

*c*) Find $\boldsymbol{\rho}^{(3)}$, the density matrix that is obtained if three qubits are selected and the remaining three are traced out.

7.4 **Codewords and nonlocality**

For the [[5,1,3]] code with stabilizer generators and logical operators as in the preceding problem,

*a*) Express $\bar{\boldsymbol{Z}}$ as a weight-3 Pauli operator, a tensor product of $\boldsymbol{I}$'s, $\boldsymbol{X}$'s, and $\boldsymbol{Z}$'s (no $\boldsymbol{Y}$'s). Note that because the code is cyclic, all cyclic permutations of your expression are equivalent ways to represent $\bar{\boldsymbol{Z}}$.

*b*) Use the Einstein locality assumption (local hidden variables) to predict a relation between the five (cyclically related) observables found in (*a*) and the observable $\boldsymbol{ZZZZZ}$. Is this relation among observables satisfied for the state $|\bar{0}\rangle$?

*c*) What would Einstein say?

7.5 **Generalized Shor code**

For integer $m \geq 2$, consider the $n = m^2$, $k = 1$ generalization of Shor's nine-qubit code, with code subspace spanned by the two states:

$$\begin{aligned} |\bar{0}\rangle &= (|000\ldots0\rangle + |111\ldots1\rangle)^{\otimes m} \ , \\ |\bar{1}\rangle &= (|000\ldots0\rangle - |111\ldots1\rangle)^{\otimes m} \ . \end{aligned} \tag{7.315}$$

*a*) Construct stabilizer generators for this code, and construct the logical operations $\bar{\boldsymbol{Z}}$ and $\bar{\boldsymbol{X}}$ such that

$$\begin{aligned} \bar{\boldsymbol{Z}}|\bar{0}\rangle = |\bar{0}\rangle \ , \quad \bar{\boldsymbol{X}}|\bar{0}\rangle = |\bar{1}\rangle \ , \\ \bar{\boldsymbol{Z}}|\bar{1}\rangle = -|\bar{1}\rangle \ , \quad \bar{\boldsymbol{X}}|\bar{1}\rangle = |\bar{0}\rangle \ . \end{aligned} \tag{7.316}$$

*b*) What is the distance of this code?

*c*) Suppose that $m$ is odd, and suppose that each of the $n = m^2$ qubits is subjected to the depolarizing channel with error probability $p$. How well does this code protect the encoded qubit? Specifically, (*i*) estimate the probability, to leading nontrivial order in $p$, of a logical bit-flip error $|\bar{0}\rangle \leftrightarrow |\bar{1}\rangle$, and (*ii*) estimate the probability, to leading nontrivial order in $p$, of a logical phase error $|\bar{0}\rangle \rightarrow |\bar{0}\rangle$, $|\bar{1}\rangle \rightarrow -|\bar{1}\rangle$.

*d*) Consider the asymptotic behavior of your answer to (*c*) for $m$ large. What condition on $p$ should be satisfied for the code to provide good protection against (*i*) bit flips and (*ii*) phase errors, in the $n \rightarrow \infty$ limit?

7.6 **Encoding circuits**

For an $[[n, k, d]]$ quantum code, an encoding transformation is a unitary $\boldsymbol{U}$ that acts as

$$\boldsymbol{U} : |\psi\rangle \otimes |0\rangle^{\otimes(n-k)} \rightarrow |\bar{\psi}\rangle \ , \tag{7.317}$$

where $|\psi\rangle$ is an arbitrary $k$-qubit state, and $|\bar{\psi}\rangle$ is the corresponding encoded state. Design a quantum circuit that implements the encoding transformation for

*a*) Shor's [[9,1,3]] code.

*b*) Steane's [[7,1,3]] code.

7.7 **Shortening a quantum code**

*a*) Consider a binary $[[n, k, d]]$ stabilizer code. Show that it is possible to choose the $n - k$ stabilizer generators so that at most two act nontrivially on the last qubit. (That is, the remaining $n - k - 2$ generators apply $\boldsymbol{I}$ to the last qubit.)

*b*) These $n-k-2$ stabilizer generators that apply $\boldsymbol{I}$ to the last qubit will still commute and are still independent if we drop the last qubit. Hence they are the generators for a code with length $n-1$ and $k+1$ encoded qubits. Show that if the original code is nondegenerate, then the distance of the shortened code is at least $d-1$. (**Hint**: First show that if there is a weight-$t$ element of the $(n-1)$-qubit Pauli group that commutes with the stabilizer of the shortened code, then there is an element of the $n$-qubit Pauli group of weight at most $t+1$ that commutes with the stabilizer of the original code.)

*c*) Apply the code-shortening procedure of (*a*) and (*b*) to the $[[5,1,3]]$ QECC. Do you recognize the code that results? (**Hint**: It may be helpful to exploit the freedom to perform a change of basis on some of the qubits.)

### 7.8 Codes for qudits

A *qudit* is a $d$-dimensional quantum system. The Pauli operators $\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ acting on qubits can be generalized to qudits as follows. Let $\{|0\rangle, |1\rangle, \ldots, |d-1\rangle\}$ denote an orthonormal basis for the Hilbert space of a single qudit. Define the operators:

$$\begin{aligned} \boldsymbol{X} &: |j\rangle \to |j+1 \pmod d\rangle \ , \\ \boldsymbol{Z} &: |j\rangle \to \omega^j |j\rangle \ , \end{aligned} \tag{7.318}$$

where $\omega = \exp(2\pi i/d)$. Then the $d \times d$ Pauli operators $\boldsymbol{E}_{r,s}$ are

$$\boldsymbol{E}_{r,s} \equiv \boldsymbol{X}^r \boldsymbol{Z}^s \ , \quad r,s = 0,1,\ldots,d-1 \tag{7.319}$$

*a*) Are the $\boldsymbol{E}_{r,s}$'s a basis for the space of operators acting on a qudit? Are they unitary? Evaluate $\mathrm{tr}(\boldsymbol{E}_{r,s}^\dagger \boldsymbol{E}_{t,u})$.

*b*) The Pauli operators obey

$$\boldsymbol{E}_{r,s}\boldsymbol{E}_{t,u} = (\eta_{r,s;t,u})\boldsymbol{E}_{t,u}\boldsymbol{E}_{r,s} \ , \tag{7.320}$$

where $\eta_{r,s;t,u}$ is a phase. Evaluate this phase.

The $n$-fold tensor products of these qudit Pauli operators form a group $G_n^{(d)}$ of order $d^{2n+1}$ (and if we mod out its $d$-element center, we obtain the group $\bar{G}_n^{(d)}$ of order $d^{2n}$). To construct a stabilizer code for qudits, we choose an abelian subgroup of $G_n^{(d)}$ with $n-k$ generators; the code is the simultaneous eigenstate with eigenvalue one of these generators. If $d$ is prime, then the code subspace has dimension $d^k$: $k$ logical qudits are encoded in a block of $n$ qudits.

*c*) Explain how the dimension might be different if $d$ is not prime. **Hint**: Consider the case $d=4$ and $n=1$.)

### 7.9 Syndrome measurement for qudits

Errors on qudits are diagnosed by measuring the stabilizer generators. For this purpose, we may invoke the two-qudit gate SUM (which generalizes the controlled-NOT), acting as

$$\mathrm{SUM} : |j\rangle \otimes |k\rangle \to |j\rangle \otimes |k+j \pmod d\rangle \ . \tag{7.321}$$

*a*) Describe a quantum circuit containing SUM gates that can be executed to measure an $n$-qudit observable of the form

$$\bigotimes_a \boldsymbol{Z}_a^{s_a} \ . \tag{7.322}$$

If $d$ is prime, then for each $r, s = 0, 1, 2, \ldots, d-1$, there is a single-qudit unitary operator $\boldsymbol{U}_{r,s}$ such that

$$\boldsymbol{U}_{r,s}\boldsymbol{E}_{r,s}\boldsymbol{U}_{r,s}^{\dagger} = \boldsymbol{Z} \ . \tag{7.323}$$

*b*) Describe a quantum circuit containing SUM gates and $\boldsymbol{U}_{r,s}$ gates that can be executed to measure an arbitrary element of $G_n^{(d)}$ of the form

$$\bigotimes_a \boldsymbol{E}_{r_a,s_a} \ . \tag{7.324}$$

7.10 **Error-detecting codes for qudits**

A qudit with $d = 3$ is called a *qutrit*. Consider a qutrit stabilizer code with length $n = 3$ and $k = 1$ encoded qutrit defined by the two stabilizer generators

$$\boldsymbol{ZZZ} \ , \quad \boldsymbol{XXX} \ . \tag{7.325}$$

*a*) Do the generators commute?

*b*) Find the distance of this code.

*c*) In terms of the orthonormal basis $\{|0\rangle, |1\rangle, |2\rangle\}$ for the qutrit, write out explicitly an orthonormal basis for the three-dimensional code subspace.

*d*) Construct the stabilizer generators for an $n = 3m$ qutrit code (where $m$ is any positive integer), with $k = n - 2$, that can detect one error.

*e*) Construct the stabilizer generators for a qudit code that detects one error, with parameters $n = d$, $k = d - 2$.

7.11 **Error-correcting code for qudits**

Consider an $n = 5$, $k = 1$ qudit stabilizer code with stabilizer generators

$$\begin{array}{ccccc} \boldsymbol{X} & \boldsymbol{Z} & \boldsymbol{Z}^{-1} & \boldsymbol{X}^{-1} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{X} & \boldsymbol{Z} & \boldsymbol{Z}^{-1} & \boldsymbol{X}^{-1} \\ \boldsymbol{X}^{-1} & \boldsymbol{I} & \boldsymbol{X} & \boldsymbol{Z} & \boldsymbol{Z}^{-1} \\ \boldsymbol{Z}^{-1} & \boldsymbol{X}^{-1} & \boldsymbol{I} & \boldsymbol{X} & \boldsymbol{Z} \end{array} \tag{7.326}$$

(the second, third, and fourth generators are obtained from the first by a cyclic permutation of the qudits).

*a*) Find the order of each generator. Are the generators really independent? Do they commute? Is the fifth cyclic permutation $\boldsymbol{Z}\ \boldsymbol{Z}^{-1}\ \boldsymbol{X}^{-1}\ \boldsymbol{I}\ \boldsymbol{X}$ independent of the rest?

*b*) Find the distance of this code. Is the code nondegenerate?

*c*) Construct the encoded operations $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{Z}}$, each expressed as an operator of weight 3. (Be sure to check that these operators obey the right commutation relations for any value of $d$.)

7.12 **Good CSS codes**

In §7.16 we derived the *quantum Gilbert-Varshamov bound*:

$$|\mathcal{E}^{(2)}| - 1 < 2^{n-k}\left(\frac{1 - 2^{-2n}}{1 - 2^{-2k}}\right). \tag{7.327}$$

This is a sufficient condition for the existence of a (possibly degenerate) binary stabilizer code that can correct all Pauli operators in a set $\mathcal{E}$; here $|\mathcal{E}^{(2)}|$ denotes the number of distinct Pauli operators of the form $E_a^{\dagger}E_b$, where $E_a, E_b \in \mathcal{E}$. One consequence of this bound is that there exist "good" $[[n, k, d = 2t+1]]$ stabilizer codes that achieve an asymptotic rate $R = k/n = 1 - H_2(2t/n) - (2t/n)\log_2 3$.

The purpose of this exercise is to show that good Calderbank-Shor-Steane (CSS) codes exist.

*a*) Derive a quantum Gilbert-Varshamov bound for CSS codes. Denote by $\mathcal{E}^X$ the set of $X$-type errors that the code can correct (those that can be expressed as a tensor product of $X$'s and $I$'s) and denote by $\mathcal{E}^Z$ the set of $Z$-type errors that the code can correct (those that can be expressed as a tensor product of $Z$'s and $I$'s). Denote by $\mathcal{E}^{X(2)}$ the set of $\{E_a^\dagger E_b\}$ where $E_a, E_b \in \mathcal{E}^X$, and similarly for $\mathcal{E}^{Z(2)}$. The quantum Gilbert-Varshamov bound for CSS codes is a sufficient condition for the existence of a CSS code with $n_X$ stabilizer generators of the $X$ type and $n_Z$ stabilizer generators of the $Z$ type that can correct all errors in $\mathcal{E}^X$ and $\mathcal{E}^Z$, expressed as an inequality involving $n_X$, $n_Z$, $|\mathcal{E}^{X(2)}|$ and $|\mathcal{E}^{Z(2)}|$.

*b*) Use the quantum Gilbert-Varshamov bound for CSS codes to show the existence of CSS codes that achieve the asymptotic rate $R = k/n = 1 - H_2(2t_X/n) - H_2(2t_Z/n)$, where the code can correct $t_X$ $X$ errors and $t_Z$ $Z$ errors.

7.13 **A cleaning lemma for CSS codes**

In §7.14 we proved the *cleaning lemma* for stabilizer codes, which says the following: For an $[[n, k]]$ stabilizer code, let $M$ denote a subset of the $n$ qubits in the code block, and let $M^c$ denote the complementary set of qubits. If $x$ is one of the code's logical Pauli operators, we say that $x$ can be *cleaned* on $M$ if there is a logically equivalent Pauli operator $x' = xy$ (where $y$ is an element of the code stabilizer $S$) such that $x'$ acts nontrivially only on $M^c$:

$$x' = I_M \otimes Q_{M^c}. \tag{7.328}$$

We say that $x$ can be *supported* on $M$ if it can be cleaned on $M^c$. Let $g(M)$ denote the number of independent logical Pauli operators that can be supported on $M$ and let $g(M^c)$ denote the number of independent Pauli operators that can be supported on $M^c$. Then the cleaning lemma asserts that

$$g(M) + g(M^c) = 2k. \tag{7.329}$$

In particular, therefore, if no logical operator can be supported on $M$, then the complete $k$-qubit logical Pauli group can be supported on its complement.

Now consider the case of an $[[n, k]]$ CSS stabilizer code, where all generators of the code stabilizer can be chosen to be either of the $X$ type (a tensor product of $X$'s and $I$'s) or the $Z$ type (a tensor product of $Z$'s and $I$'s); furthermore, the generators of the logical Pauli group can also be chosen to be either $X$ type or $Z$ type. Let $g^X(M)$ denote the number of independent $X$-type logical Pauli operators supported on $M$, and let $g^Z(M^c)$ denote the number of independent $Z$-type logical Pauli operators supported on $M^c$. Show that

$$g^X(M) + g^Z(M^c) = k. \tag{7.330}$$

It follows that if no $X$-type logical Pauli operators can be supported on $M$, then all $Z$-type logical operators can be supported on its complement.

7.14 Bound on $D$-dimensional codes

In class (and in the notes) we proved a bound on $[[n, k, d]]$ for local stabilizer codes in 2 dimensions: $kd^2 = O(n)$, where $k$ is the number of encoded qubits, $d$ is the code distance, and $n$ is the code length.

Using similar reasoning, derive a bound on $D$-dimensional stabilizer codes of the form $kd^\alpha = O(n)$. Express $\alpha$ in terms of $D$.

7.15 **Price of a code**

The distance $d$ of a stabilizer code is the size of the smallest set of qubits in the code block that supports a nontrivial logical operator. In contrast, the *price* $p$ of a code is defined as the size of the smallest set of qubits that supports *all* of the code's logical operators. Evidently $p \geq d$.

*a*) What is the price of the [[7,1,3]] quantum code?

*b*) Use the cleaning lemma to show that $p \leq n - d + 1$.

*c*) Show that $p \geq k + d - 1$. **Hint**: Use this result proved in class (and in the notes): Suppose that the code block can be divided into three parts $ABC$ such that $A$ and $B$ are both correctable. Then $k \leq |C|$, where $|C|$ is the number of qubits in $C$.

*d*) For an $[[n, k, d]]$ stabilizer code, prove the *quantum Singleton bound*: $n - k \geq 2(d - 1)$.

7.16 **Price of a local code**

Using the holographic principle and the union lemma (described in class and in the notes), derive a bound on the price $p$ of a $D$-dimensional stabilizer code of the form $pd^\beta = O(n)$. Express $\beta$ in terms of $D$.

7.17 **Higher-dimensional toric codes**

To understand the logical operators of the 2D toric code, we found it convenient to use the concept of a *chain complex*. We defined vector spaces $V_0, V_1, V_2$ over $\mathbb{F}^2$ and boundary operators $\partial_1, \partial_2$ such that

$$V_2 \xrightarrow{\partial_2} V_1 \xrightarrow{\partial_1} V_0 \tag{7.331}$$

and

$$\partial_1 \circ \partial_2 = 0. \tag{7.332}$$

For example, for the purpose of formulating the $\boldsymbol{Z}$-type logical operators of the toric code, vectors in $V_2$ are *2-chains*, which assign a bit to each lattice plaquette, vectors in $V_1$ are *1-chains*, which assign a bit to each edge, and vectors in $V_0$ are *0-chains*, which assign a bit to each site. We saw that a 2-chain $\omega$ may be interpreted as a stabilizer element (a product of $\boldsymbol{Z}$-type plaquette operators), a 1-chain $\epsilon$ may be interpreted as a $\boldsymbol{Z}$-type error operator, and the 0-chain $\sigma = \partial_1 \epsilon$ may be interpreted as the syndrome of $\epsilon$ (determined by measuring the $\boldsymbol{X}$-type site operators). The property $\partial_1 \circ \partial_2 = 0$ expresses that a $\boldsymbol{Z}$-type stabilizer operator has a trivial syndrome.

A $\boldsymbol{Z}$-type operator which lies in the code stabilizer $S$ is the boundary $\partial_2 \omega$ of some 2-chain $\omega$; it is contained in the image of $\partial_2$. A $\boldsymbol{Z}$-type operator $\epsilon$ which lies in the normalizer $S^\perp$ (commutes with the stabilizer) is a *1-cycle* with trivial boundary, $\partial_1 \epsilon = 0$; it is contained in the kernel of $\partial_1$. Thus the code's $\boldsymbol{Z}$-type logical Pauli operators are elements of the coset space. $\mathrm{Ker}(\partial_1)/\mathrm{Im}(\partial_2)$.

We have also seen how this chain-complex language can be applied to any CSS stabilizer code. In this problem, we'll apply it to toric codes in more than 2 dimensions.

Consider a 3D version of the toric code defined on an $L \times L \times L$ cubic lattice with periodic boundary conditions. Qubits reside on lattice links. There is a weight-4

$\boldsymbol{Z}$-type stabilizer generator associated with each lattice plaquette, and a weight-6 $\boldsymbol{X}$-type stabilizer generator associated with each lattice site.

*a*) Construct a chain complex suited for describing the $\boldsymbol{Z}$-type logical operators of this code, and find the logical operators. How many encoded qubits are there? What is the minimal weight $d_Z$ of a $\boldsymbol{Z}$-type logical operator?

*b*) Construct a chain complex suited for describing the $\boldsymbol{X}$-type logical operators of this code, and find the logical operators. Check that these $\boldsymbol{X}$-type logical operators have the expected commutation relations with the $\boldsymbol{Z}$-type logical operators. What is the minimal weight $d_X$ of an $\boldsymbol{X}$-type logical operator?

Consider a 4D version of the toric code defined on an $L \times L \times L \times L$ hypercubic lattice with periodic boundary conditions. Qubits reside on lattice plaquettes. There is a weight-6 $\boldsymbol{Z}$-type stabilizer generator associated with each lattice cube, and a weight-6 $\boldsymbol{X}$-type stabilizer generator associated with each lattice link.

*c*) Construct a chain complex suited for describing the $\boldsymbol{Z}$-type logical operators of this code, and find the logical operators. How many encoded qubits are there? What is the minimal weight $d_Z$ of a $\boldsymbol{Z}$-type logical operator?

*d*) Construct a chain complex suited for describing the $\boldsymbol{X}$-type logical operators of this code, and find the logical operators. Check that these $\boldsymbol{X}$-type logical operators have the expected commutation relations with the $\boldsymbol{Z}$-type logical operators. What is the minimal weight $d_X$ of an $\boldsymbol{X}$-type logical operator?