Chapter 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

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

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

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

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 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:

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

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:

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

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 $4p^2$, and the probability of a bit flip error is bounded above by $12p^2$. 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 $p < 1/16$.

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 Criteria 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.

To begin with, consider a single qubit, initially in a pure state, that interacts with its environment in an arbitrary manner. We know from Chapter

3 that there is no loss or generality (we may still represent the most general superoperator acting on our qubit) if we assume that the initial state of the environment is a pure state, which we will denote as $|0\rangle_E$. Then the evolution of the qubit and its environment can be described by a unitary transformation

$$\begin{aligned} \boldsymbol{U}: \quad & |0\rangle \otimes |0\rangle_E \rightarrow |0\rangle \otimes |e_{00}\rangle_E + |1\rangle \otimes |e_{01}\rangle_E \ , \\ & |1\rangle \otimes |0\rangle_E \rightarrow |0\rangle \otimes |e_{10}\rangle_E + |1\rangle \otimes |e_{11}\rangle_E \ ; \end{aligned} \tag{7.9}$$

here the four $|e_{ij}\rangle_E$ are states of the environment that need not be normalized or mutually orthogonal (though they do satisfy some constraints that follow from the unitarity of $\boldsymbol{U}$). Under $\boldsymbol{U}$, an arbitrary state $|\psi\rangle = a|0\rangle + b|1\rangle$ of the qubit evolves as

$$\begin{aligned} \boldsymbol{U}: \ (a|0\rangle + b|1\rangle)|0\rangle_E \rightarrow & \ a(|0\rangle|e_{00}\rangle_E + |1\rangle|e_{01}\rangle_E) \\ & + b(|0\rangle|e_{10}\rangle_E + |1\rangle|e_{11}\rangle_E) \end{aligned}$$

$$\begin{aligned} &= (a|0\rangle + b|1\rangle) \otimes \frac{1}{2}(|e_{00}\rangle_E + |e_{11}\rangle_E) \\ &+ (a|0\rangle - b|1\rangle) \otimes \frac{1}{2}(|e_{00}\rangle_E - |e_{11}\rangle_E) \\ &+ (a|1\rangle + b|0\rangle) \otimes \frac{1}{2}(|e_{01}\rangle_E + |e_{10}\rangle_E) \\ &+ (a|1\rangle - b|0\rangle) \otimes \frac{1}{2}(|e_{01}\rangle_E - |e_{10}\rangle_E) \end{aligned}$$

$$\begin{aligned} \equiv \boldsymbol{I}|\psi\rangle \otimes |e_I\rangle_E + \boldsymbol{X}|\psi\rangle \otimes |e_X\rangle_E + \boldsymbol{Y}|\psi\rangle \otimes |e_Y\rangle_E & \\ + \boldsymbol{Z}|\psi\rangle \otimes |e_Z\rangle_E. & \end{aligned} \tag{7.10}$$

The action of $\boldsymbol{U}$ can be expanded in terms of the (unitary) Pauli operators $\{\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}$, simply because these are a basis for the vector space of 2×2 matrices. Heuristically, we might interpret this expansion by saying that one of four possible things happens to the qubit: nothing ($\boldsymbol{I}$), a bit flip ($\boldsymbol{X}$), a phase flip ($\boldsymbol{Z}$), or both ($\boldsymbol{Y} = i\boldsymbol{X}\boldsymbol{Z}$). However, this classification should not be taken literally, because unless the states $\{|e_I\rangle, |e_X\rangle, |e_Y\rangle, |e_Z\rangle\}$ of the environment are all mutually orthogonal, there is no conceivable measurement that could perfectly distinguish among the four alternatives.

Similarly, an arbitrary $2^n \times 2^n$ matrix acting on an n-qubit Hilbert space can be expanded in terms of the 2^{2n} operators

$$\{\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}^{\otimes n}; \tag{7.11}$$

that is, each such operator can be expressed as a tensor-product "string" of single-qubit operators, with each operator in the string chosen from among the identity and the three Pauli matrices $\boldsymbol{X}, \boldsymbol{Y}$, and $\boldsymbol{Z}$. Thus, the action of an arbitrary unitary operator on n qubits plus their environment can be expanded as

$$|\psi\rangle \otimes |0\rangle_E \rightarrow \sum_a \boldsymbol{E}_a|\psi\rangle \otimes |e_a\rangle_E; \tag{7.12}$$

here the index a ranges over 2^{2n} values. The $\{\boldsymbol{E}_a\}$ are the linearly independent Pauli operators acting on the n qubits, and the $\{|e_a\rangle_E\}$ are the corresponding states of the environment (which are *not* assumed to be normalized or mutually orthogonal). A crucial feature of this expansion for what follows is that each $\boldsymbol{E}_a$ is a unitary operator.

Eq. (7.12) provides the conceptual foundation of quantum error correction. In devising a quantum error-correcting code, we identify a subset $\mathcal{E}$ of all the Pauli operators,

$$\mathcal{E} \subseteq \{\boldsymbol{E}_a\} \equiv \{\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}^{\otimes n} \; ; \tag{7.13}$$

these are the errors that we wish to be able to correct. Our aim will be to perform a collective measurement of the n qubits in the code block that will enable us to diagnose which error $\boldsymbol{E}_a \in \mathcal{E}$ occurred. If $|\psi\rangle$ is a state in the code subspace, then for some (but not all) codes this measurement will prepare a state $\boldsymbol{E}_a|\psi\rangle \otimes |e_a\rangle_E$, where the value of a is known from the measurement outcome. Since $\boldsymbol{E}_a$ is unitary, we may proceed to apply $\boldsymbol{E}_a^\dagger (= \boldsymbol{E}_a)$ to the code block, thus recovering the undamaged state $|\psi\rangle$.

Each Pauli operator can be assigned a *weight*, an integer t with $0 \leq t \leq n$; the weight is the number of qubits acted on by a nontrivial Pauli matrix ($\boldsymbol{X}, \boldsymbol{Y}$, or $\boldsymbol{Z}$). Heuristically, then, we can interpret a term in the expansion eq. (7.12) where $\boldsymbol{E}_a$ has weight t as an event in which errors occur on t qubits (but again we cannot take this interpretation too literally if the states $\{|e_a\rangle_E\}$ are not mutually orthogonal). Typically, we will take $\mathcal{E}$ to be the set of all Pauli operators of weight up to and including t; then if we can recover from any error superoperator with support on the set $\mathcal{E}$, we will say that the code can correct t errors. In adopting such an error set, we are implicitly assuming that the errors afflicting different qubits are only weakly correlated with one another, so that the amplitude for more than t errors on the n qubits is relatively small.

Given the set $\mathcal{E}$ of errors that are to be corrected, what are the necessary and sufficient conditions to be satisfied by the code subspace in order that recovery is possible? Let us denote by $\{\ |\bar{i}\rangle\ \}$ an orthonormal basis for the code subspace. (We will refer to these basis elements as "codewords".) It will clearly be *necessary* that

$$\langle\bar{j}|\boldsymbol{E}_b^\dagger\boldsymbol{E}_a|\bar{i}\rangle = 0, \quad i \neq j, \tag{7.14}$$

where $\boldsymbol{E}_{a,b} \in \mathcal{E}$. If this condition were not satisfied for some $i \neq j$, then errors would be able to destroy the perfect distinguishability of orthogonal codewords, and encoded quantum information could surely be damaged. (A more explicit derivation of this necessary condition will be presented below.) We can also easily see that a *sufficient* condition is

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

In this case the $\boldsymbol{E}_a$'s take the code subspace to a set of mutually orthogonal "error subspaces"

$$\mathcal{H}_a = \boldsymbol{E}_a\mathcal{H}_{code}. \tag{7.16}$$

Suppose, then that an arbitrary state $|\psi\rangle$ in the code subspace is prepared, and subjected to an error. The resulting state of code block and environment is

$$\sum_{\boldsymbol{E}_a\in\mathcal{E}} \boldsymbol{E}_a|\psi\rangle \otimes |e_a\rangle_E, \tag{7.17}$$

where the sum is restricted to the errors in the set $\mathcal{E}$. We may then perform an orthogonal measurement that projects the code block onto one of the spaces $\mathcal{H}_a$, so that the state becomes

$$\boldsymbol{E}_a|\psi\rangle \otimes |e_a\rangle_E. \tag{7.18}$$

We finally apply the unitary operator $\boldsymbol{E}_a^\dagger$ to the code block to complete the recovery procedure.

A code that satisfies the condition eq. (7.15) is called a *nondegenerate* code. This terminology signifies that there is a measurement that can unambiguously diagnose the error $\boldsymbol{E}_a \in \mathcal{E}$ that occurred. But the example of the nine-qubit code has already taught us that more general codes are possible. The nine-qubit code is *degenerate*, because phase errors acting on different qubits in the same cluster of three affect the code subspace in precisely the same way (*e.g.*, $\boldsymbol{Z}_1|\psi\rangle = \boldsymbol{Z}_2|\psi\rangle$). Though no measurement can determine which qubit suffered the error, this need not pose an obstacle to successful recovery.

The necessary and sufficient condition for recovery to be possible is easily stated:

$$\langle\bar{j}|\boldsymbol{E}_b^\dagger \boldsymbol{E}_a|\bar{i}\rangle = C_{ba}\delta_{ij}, \tag{7.19}$$

where $\boldsymbol{E}_{a,b} \in \mathcal{E}$, and $C_{ba} = \langle\bar{i}|\boldsymbol{E}_b^\dagger \boldsymbol{E}_a|\bar{i}\rangle$ is an arbitrary Hermitian matrix. The nontrivial content of this condition that goes beyond the weaker necessary condition eq. (7.14) is that $\langle\bar{i}|\boldsymbol{E}_b^\dagger \boldsymbol{E}_a|\bar{i}\rangle$ does not depend on i. The origin of this condition is readily understood — were it otherwise, in identifying an error subspace $\mathcal{H}_a$ we would acquire some information about the encoded state, and so would inevitably disturb that state.

To prove that the condition eq. (7.19) is necessary and sufficient, we invoke the theory of superoperators developed in Chapter 3. Errors acting on the code block are described by a superoperator, and the issue is whether another superoperator (the recovery procedure) can be constructed that will reverse the effect of the error. In fact, we learned in Chapter 3 that the only superoperators that can be inverted are unitary operators. But now we are demanding a bit less. We are not required to be able to reverse the action of the error superoperator on any state in the n-qubit code block; rather, it is enough to be able to reverse the errors when the initial state resides in the k-qubit encoded subspace.

An alternative way to express the action of an error on one of the code basis states $|\bar{i}\rangle$ (and the environment) is

$$|\bar{i}\rangle \otimes |0\rangle_E \to \sum_\mu \boldsymbol{M}_\mu|\bar{i}\rangle \otimes |\mu\rangle_E, \tag{7.20}$$

where now the states $|\mu\rangle_E$ are elements of an *orthonormal basis* for the environment, and the matrices $\boldsymbol{M}_\mu$ are linear combinations of the Pauli operators

$\boldsymbol{E}_a$ contained in $\mathcal{E}$, satisfying the operator-sum normalization condition

$$\sum_\mu \boldsymbol{M}_\mu^\dagger \boldsymbol{M}_\mu = \boldsymbol{I}. \tag{7.21}$$

The error can be reversed by a recovery superoperator if there exist operators $\boldsymbol{R}_\nu$ such that

$$\sum_\nu \boldsymbol{R}_\nu^\dagger \boldsymbol{R}_\nu = \boldsymbol{I}, \tag{7.22}$$

and

$$\sum_{\mu,\nu} \boldsymbol{R}_\nu \boldsymbol{M}_\mu |\bar{i}\rangle \otimes |\mu\rangle_E \otimes |\nu\rangle_A$$

$$= |\bar{i}\rangle \otimes |\text{stuff}\rangle_{EA}; \tag{7.23}$$

here the $|\nu\rangle_A$'s are elements of an orthonormal basis for the Hilbert space of the *ancilla* that is employed to implement the recovery operation, and the state $|\text{stuff}\rangle_{EA}$ of environment and ancilla must not depend on i. It follows that

$$\boldsymbol{R}_\nu \boldsymbol{M}_\mu |\bar{i}\rangle = \lambda_{\nu\mu} |\bar{i}\rangle; \tag{7.24}$$

for each μ and ν; the product $\boldsymbol{R}_\nu \boldsymbol{M}_\mu$ acting on the code subspace is a multiple of the identity. Using the normalization condition satisfied by the $\boldsymbol{R}_\nu$'s, we infer that

$$\boldsymbol{M}_\delta^\dagger \boldsymbol{M}_\mu |\bar{i}\rangle = \boldsymbol{M}_\delta^\dagger \left(\sum_\nu \boldsymbol{R}_\nu^\dagger \boldsymbol{R}_\nu \right) \boldsymbol{M}_\mu |\bar{i}\rangle = \sum_\nu \lambda_{\nu\delta}^* \lambda_{\nu\mu} |\bar{i}\rangle, \tag{7.25}$$

so that $\boldsymbol{M}_\delta^\dagger \boldsymbol{M}_\mu$ is likewise a multiple of the identity acting on the code subspace. In other words

$$\langle \bar{j} | \boldsymbol{M}_\delta^\dagger \boldsymbol{M}_\mu | \bar{i} \rangle = C_{\delta\mu} \delta_{ij}; \tag{7.26}$$

since each $\boldsymbol{E}_a$ in $\mathcal{E}$ is a linear combination of $\boldsymbol{M}_\mu$'s, eq. (7.19) then follows.

Another instructive way to understand why eq. (7.26) is a necessary condition for error recovery is to note that if the code block is prepared in the state $|\psi\rangle$, and an error acts according to eq. (7.20), then the density matrix for the environment that we obtain by tracing over the code block is

$$\rho_E = \sum_{\mu,\nu} |\mu\rangle_E \langle \psi | \boldsymbol{M}_\nu^\dagger \boldsymbol{M}_\mu | \psi \rangle_E \langle \nu |. \tag{7.27}$$

Error recovery can proceed successfully only if there is no way to acquire any information about the state $|\psi\rangle$ by performing a measurement on the environment. Therefore, we require that ρ_E be independent of $|\psi\rangle$, if $|\psi\rangle$ is any state in the code subspace; eq. (7.26) then follows.

To see that eq. (7.26) is sufficient for recovery as well as necessary, we can explicitly construct the superoperator that reverses the error. For this purpose it is convenient to choose our basis $\{|\mu\rangle_E\}$ for the environment so that the matrix $C_{\delta\mu}$ in eq. (7.26) is diagonalized:

$$\langle\bar{j}|\boldsymbol{M}_\delta^\dagger \boldsymbol{M}_\mu|\bar{i}\rangle = C_\mu \delta_{\delta\mu}\delta_{ij} \ , \tag{7.28}$$

where $\sum_\mu C_\mu = 1$ follows from the operator-sum normalization condition. For each ν with $C_\nu \neq 0$, let

$$\boldsymbol{R}_\nu = \frac{1}{\sqrt{C_\nu}} \sum_i |\bar{i}\rangle\langle\bar{i}|\boldsymbol{M}_\nu^\dagger, \tag{7.29}$$

so that $\boldsymbol{R}_\nu$ acts according to

$$\boldsymbol{R}_\nu : \boldsymbol{M}_\mu|\bar{i}\rangle \to \sqrt{C_\nu}\delta_{\mu\nu}|\bar{i}\rangle. \tag{7.30}$$

Then we easily see that

$$\begin{aligned}&\sum_{\mu,\nu} \boldsymbol{R}_\nu \boldsymbol{M}_\mu |\bar{i}\rangle \otimes |\mu\rangle_E \otimes |\nu\rangle_A \\ &= |\bar{i}\rangle \otimes (\sum_\nu \sqrt{C_\nu}|\nu\rangle_E \otimes |\nu\rangle_A);\end{aligned} \tag{7.31}$$

the superoperator defined by the $\boldsymbol{R}_\nu$'s does indeed reverse the error. It only remains to check that the $\boldsymbol{R}_\nu$'s satisfy the normalization condition. We have

$$\sum_\nu \boldsymbol{R}_\nu^\dagger \boldsymbol{R}_\nu = \sum_{\nu,i} \frac{1}{C_\nu} \sum_\nu \boldsymbol{M}_\nu |\bar{i}\rangle\langle\bar{i}|\boldsymbol{M}_\nu^\dagger \ , \tag{7.32}$$

which is the orthogonal projection onto the space of states that can be reached by errors acting on codewords. Thus we can complete the specification of the recovery superoperator by adding one more element to the operator sum — the projection onto the complementary subspace.

In brief, eq. (7.19) is a sufficient condition for error recovery because it is possible to choose a basis for the error operators (not necessarily the Pauli operator basis) that diagonalizes the matrix C_{ab}, and in this basis we can unambiguously diagnose the error by performing a suitable orthogonal measurement. (The eigenmodes of C_{ab} with eigenvalue zero, like $\boldsymbol{Z}_1 - \boldsymbol{Z}_2$ in the case of the 9-qubit code, correspond to errors that occur with probability zero.) We see that, once the set $\mathcal{E}$ of possible errors is specified, the recovery operation is determined. In particular, no information is needed about the states $|e_a\rangle_E$ of the environment that are associated with the errors $\boldsymbol{E}_a$. Therefore, the code works equally effectively to control unitary errors or decoherence errors (as long as the amplitude for errors outside of the set $\mathcal{E}$ is negligible). Of course, in the case of a nondegenerate code, C_{ab} is already diagonal in the Pauli basis, and we can express the recovery basis as

$$\boldsymbol{R}_a = \sum_i |\bar{i}\rangle\langle\bar{i}|\boldsymbol{E}_a^\dagger \ ; \tag{7.33}$$

there is an $\boldsymbol{R}_a$ corresponding to each $\boldsymbol{E}_a$ in $\mathcal{E}$.

We have described error correction as a two step procedure: first a collective measurement is conducted to diagnose the error, and secondly, based on the measurement outcome, a unitary transformation is applied to reverse the error. This point of view has many virtues. In particular, it is the quantum measurement procedure that seems to enable us to tame a continuum of possible errors, as the measurement projects the damaged state into one of a discrete set of outcomes, for each of which there is a prescription for recovery. But in fact measurement is not an essential ingredient of quantum error correction. The recovery superoperator of eq. (7.31) may of course be viewed as a unitary transformation acting on the code block and an ancilla. This superoperator can describe a measurement followed by a unitary operator if we imagine that the ancilla is subjected to an orthogonal measurement, but the measurement is not necessary.

If there is no measurement, we are led to a different perspective on the reversal of decoherence achieved in the recovery step. When the code block interacts with its environment, it becomes entangled with the environment, and the Von Neumann entropy of the environment increases (as does the entropy of the code block). If we are unable to control the environment, that increase in its entropy can never be reversed; how then, is quantum error correction possible? The answer provided by eq. (7.31) is that we may apply a unitary transformation to the data and to an ancilla that we *do* control. If the criteria for quantum error correction are satisfied, this unitary can be chosen to transform the entanglement of the data with the environment into entanglement of ancilla with environment, restoring the purity of the data in the process, as in:

– Figure –

While measurement is not a necessary part of error correction, the ancilla is absolutely essential. The ancilla serves as a depository for the entropy inserted into the code block by the errors — it "heats" as the data "cools." If we are to continue to protect quantum information stored in quantum memory for a long time, a continuous supply of ancilla qubits should be provided that can be discarded after use. Alternatively, if the ancilla is to be recycled, it must first be erased. As discussed in Chapter 1, the erasure is dissipative and requires the expenditure of power. Thus principles of thermodynamics dictate that we cannot implement (quantum) error correction for free. Errors cause entropy to seep into the data. This entropy can be transferred to the ancilla by means of a reversible process, but work is needed to pump entropy from the ancilla back to the environment.

7.3 Some General Properties of QECC's

7.3.1 Distance

A quantum code is said to be *binary* if it can be represented in terms of qubits. In a binary code, a code subspace of dimension 2^k is embedded in a space of dimension 2^n, where k and $n > k$ are integers. There is actually no need to require that the dimensions of these spaces be powers of two (see the exercises); nevertheless we will mostly confine our attention here to binary coding, which is the simplest case.

In addition to the block size 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 a Pauli operator $\boldsymbol{E}$ such that

$$\langle \bar{i}|\boldsymbol{E}_a|\bar{j}\rangle \neq C_a\delta_{ij}. \tag{7.34}$$

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 an QECC can correct t errors if the set $\mathcal{E}$ of $\boldsymbol{E}_a$'s that allow recovery includes all Pauli operators of weigh t or less. Our definition of distance implies that the criterion for error correction

$$\langle\bar{i}|\boldsymbol{E}_a^\dagger \boldsymbol{E}_b|\bar{j}\rangle = C_{ab}\delta_{ij}, \tag{7.35}$$

will be satisfied by all Pauli operators $\boldsymbol{E}_{a,b}$ of weight t or less, provided that $d \geq 2t+1$. Therefore, a QECC with distance $d = 2t+1$ can correct t errors.

7.3.2 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 are especially likely to have suffered errors. Perhaps we saw a hammer strike those qubits. Or perhaps you sent a block of n qubits to me, but $t < n$ of the qubits were lost and never received. I am confident that the $n-t$ qubits that did arrive were well packaged and were received undamaged. But I replace the t missing qubits with the (arbitrarily chosen) state $|00\ldots0\rangle$, realizing full well that these qubits are likely to be in error.

A given code can protect against more errors if the errors occur at known locations instead of unknown locations. In fact, 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 $\boldsymbol{E}_a^\dagger \boldsymbol{E}_b$ 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 $t+1$.

In particular, a QECC that corrects t errors in arbitrary locations can correct $2t$ errors in known locations.

7.3.3 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 superoperator 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.

Such a code has the property that

$$\langle \bar{i} | \boldsymbol{E}_a | \bar{j} \rangle = C_a \delta_{ij} \tag{7.36}$$

for every Pauli operator $\boldsymbol{E}_a$ of weight t or less, or

$$\boldsymbol{E}_a | \bar{i} \rangle = C_a | \bar{i} \rangle + | \varphi^{\perp}_{ai} \rangle \ , \tag{7.37}$$

where $|\varphi^{\perp}_{ai}\rangle$ is an unnormalized vector orthogonal to the code subspace. Therefore, the action on a state $|\psi\rangle$ in the code subspace of an error superoperator with support on $\mathcal{E}$ is

$$|\psi\rangle \otimes |0\rangle_E \to \sum_{\boldsymbol{E}_a \in \mathcal{E}} \boldsymbol{E}_a |\psi\rangle \otimes |e_a\rangle_E = |\psi\rangle \otimes \left(\sum_{\boldsymbol{E}_a \in \mathcal{E}} C_a |e_a\rangle_E \right) + |\text{orthog}\rangle \ , \tag{7.38}$$

where $|\text{orthog}\rangle$ denotes a vector orthogonal to the code subspace.

Now we can perform a "fuzzy" orthogonal measurement on the data, with two outcomes: the state is projected onto either the code subspace or the complementary subspace. If the first outcome is obtained, the undamaged state $|\psi\rangle$ is recovered. If the second outcome is found, an error has been detected. We conclude that our QECC with distance d can detect $d - 1$ errors. In particular, then, a QECC that can correct t errors can detect $2t$ errors.

7.3.4 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

$$\boldsymbol{\rho}^{(t)} = \mathrm{tr}_{(n-t)}|\psi\rangle\langle\psi| \ , \tag{7.39}$$

the density matrix of the t qubits. Then this density matrix is totally random:

$$\boldsymbol{\rho}^{(t)} = \frac{1}{2^t}\boldsymbol{I}; \tag{7.40}$$

(In any distance-$(t+1)$ code, 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 only if the code is nondegenerate will the density matrix of the t qubits be a multiple of the identity.)

To verify the property eq. (7.40), we note that for a nondegenerate distance-$(t+1)$ code,

$$\langle\bar{i}|\boldsymbol{E}_a|\bar{j}\rangle = 0 \tag{7.41}$$

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

$$\mathrm{tr}(\boldsymbol{\rho}^{(t)}\boldsymbol{E}_a) = 0, \tag{7.42}$$

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:

$$\boldsymbol{\rho}^{(t)} = \left(\frac{1}{2^t}\right)\boldsymbol{I} + \sum_{\boldsymbol{E}_a \neq \boldsymbol{I}} \rho_a \boldsymbol{E}_a \ . \tag{7.43}$$

Since the $\boldsymbol{E}_a$'s satisfy

$$\left(\frac{1}{2^t}\right)\mathrm{tr}(\boldsymbol{E}_a\boldsymbol{E}_b) = \delta_{ab} \ , \tag{7.44}$$

we find that each $\rho_a = 0$, and we conclude that $\boldsymbol{\rho}^{(t)}$ is a multiple of the identity.

7.4 Probability of Failure

7.4.1 Fidelity bound

If the support of the error superoperator 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 superoperator 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

$$\begin{aligned} & |\psi\rangle \otimes |0\rangle_E \to \sum_a \boldsymbol{E}_a|\psi\rangle \otimes |e_a\rangle_E \\ \equiv \quad & \sum_{\boldsymbol{E}_a\in\mathcal{E}} \boldsymbol{E}_a|\psi\rangle \otimes |e_a\rangle_E + \sum_{\boldsymbol{E}_b\notin\mathcal{E}} \boldsymbol{E}_b|\psi\rangle \otimes |e_b\rangle_E \\ \equiv \quad & |\mathrm{GOOD}\rangle + |\mathrm{BAD}\rangle \ . \end{aligned} \tag{7.45}$$

The recovery operation (a unitary acting on the data and the ancilla) 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

$$|\mathrm{GOOD}'\rangle + |\mathrm{BAD}'\rangle \ ; \tag{7.46}$$

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

$$|\mathrm{GOOD}'\rangle = |\psi\rangle \otimes |s\rangle_{EA} \ , \tag{7.47}$$

where $|s\rangle_{EA}$ is some state of the environment and ancilla.

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

$$\langle e_a|e_b\rangle = 0 \quad \text{for} \quad \boldsymbol{E}_a \in \mathcal{E}, \ \boldsymbol{E}_b \notin \mathcal{E}. \tag{7.48}$$

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

$$F = \langle\psi|\boldsymbol{\rho}_{\mathrm{rec}}|\psi\rangle \tag{7.49}$$

be its fidelity. Now, since $|\mathrm{BAD}'\rangle$ is orthogonal to $|\mathrm{GOOD}'\rangle$ (that is, $|\mathrm{BAD}'\rangle$ has no component along $|\psi\rangle|s\rangle_{EA}$), the fidelity will be

$$F = \langle\psi|\boldsymbol{\rho}_{\mathrm{GOOD}'}|\psi\rangle + \langle\psi|\boldsymbol{\rho}_{\mathrm{BAD}'}|\psi\rangle \ , \tag{7.50}$$

where

$$\boldsymbol{\rho}_{\mathrm{GOOD}'} = \mathrm{tr}_{EA}\left(|\mathrm{GOOD}'\rangle\langle\mathrm{GOOD}'|\right) \ , \quad \boldsymbol{\rho}_{\mathrm{BAD}'} = \mathrm{tr}_{EA}\left(|\mathrm{BAD}'\rangle\langle\mathrm{BAD}'|\right) \ . \tag{7.51}$$

The fidelity of the recovered state therefore satisfies

$$F \geq \ \langle\psi|\boldsymbol{\rho}_{\mathrm{GOOD}'}|\psi\rangle = \parallel |s\rangle_{EA} \parallel^2 = \parallel |\mathrm{GOOD}'\rangle \parallel^2 \ . \tag{7.52}$$

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

$$F \geq \parallel |\mathrm{GOOD}\rangle \parallel^2 = \parallel \sum_{\boldsymbol{E}_a\in\mathcal{E}} \boldsymbol{E}_a|\psi\rangle \otimes |e_a\rangle_E \parallel^2 \ . \tag{7.53}$$

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

$$|\mathrm{BAD}'\rangle = |\mathrm{BAD}'_{\parallel}\rangle + |\mathrm{BAD}'_{\perp}\rangle \tag{7.54}$$

Then reasoning just as above we obtain

$$F \geq \parallel |\mathrm{GOOD}'\rangle + |\mathrm{BAD}'_{\parallel}\rangle \parallel^2 \tag{7.55}$$

Of course, since both the error operation and the recovery operation are unitary acting on data, environment, and ancilla, the complete state $|\mathrm{GOOD}'\rangle + |\mathrm{BAD}'\rangle$ is normalized, or

$$\parallel |\mathrm{GOOD}'\rangle + |\mathrm{BAD}'_{\parallel}\rangle \parallel^2 + \parallel |\mathrm{BAD}'_{\perp}\rangle \parallel^2 = 1 \ , \tag{7.56}$$

and eq. (7.55) becomes

$$F \geq \ 1 - \parallel |\mathrm{BAD}'_{\perp}\rangle \parallel^2 \ . \tag{7.57}$$

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

$$1 - F \leq \parallel |\mathrm{BAD}'\rangle \parallel^2 = \parallel |\mathrm{BAD}\rangle \parallel^2 \equiv \parallel \sum_{\boldsymbol{E}_b \notin \mathcal{E}} \boldsymbol{E}_b |\psi\rangle \otimes |e_b\rangle_E \parallel^2 \ . \tag{7.58}$$

This is our general bound on the "failure probability" of the recovery operation. The result eq. (7.53) then follows in the special case where $|\mathrm{GOOD}\rangle$ and $|\mathrm{BAD}\rangle$ are orthogonal states.

7.4.2 Uncorrelated errors

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 superoperator is a tensor product of single-qubit superoperators. If in fact the errors act on all the qubits in the same way, we can express the n-qubit superoperator as

$$\$^{(n)}_{\text{error}} = \left[\$^{(1)}_{\text{error}}\right]^{\otimes n} \ , \tag{7.59}$$

where $\$^{(1)}_{\text{error}}$ is a one-qubit superoperator whose action (in its unitary representation) has the form

$$|\psi\rangle \otimes |0\rangle_E \to |\psi\rangle \otimes |e_I\rangle_E + \boldsymbol{X}|\psi\rangle \otimes \ |e_X\rangle_E + \boldsymbol{Y}|\psi\rangle \otimes |e_Y\rangle_E \\ + \boldsymbol{Z}|\psi\rangle \otimes |e_Z\rangle_E \ . \tag{7.60}$$

The effect of the errors on encoded information is especially easy to analyze if we suppose further that each of the three states of the environment $|e_{X,Y,Z}\rangle$ is orthogonal to the state $|e_I\rangle$. 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 p_{error} for each qubit, where

$$\langle e_I | e_I \rangle = 1 - p_{\text{error}} \ . \tag{7.61}$$

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

$$1 - F \ \leq \ \sum_{s=t+1}^{n} \binom{n}{s} p^s_{\text{error}} \left(1 - p_{\text{error}}\right)^{n-s} \ \leq \ \binom{n}{t+1} p^{t+1}_{\text{error}} \ . \tag{7.62}$$

(For each of the $\binom{n}{t+1}$ ways of choosing $t+1$ locations, the probability that errors occurs at every one of those locations is $p_{\rm error}^{t+1}$, where we disregard whether additional errors occur at the remaining $n-t-1$ locations. Therefore, the final expression in eq. (7.62) is an upper bound on the probability that at least $t+1$ errors occur in the block of n qubits.) For $p_{\rm error}$ small and t large, the fidelity of the encoded data is a substantial improvement over the fidelity $F = 1 - O(p)$ maintained by an unprotected qubit.

For a general error superoperator 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 there is no decoherence at all — the "errors" arise because unknown unitary transformations act on the qubits. (If the unitary transformation $\boldsymbol{U}$ acting on a qubit were known, we could recover from the "error" simply by applying $\boldsymbol{U}^\dagger$.)

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

$$\boldsymbol{U}^{(1)} = \sqrt{1-p} + i\sqrt{p}\, \boldsymbol{W}, \tag{7.63}$$

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. (7.63) occurs, the fidelity of the resulting state is

$$F = \left|\langle\psi|\boldsymbol{U}^{(1)}|\psi\rangle\right|^2 = 1 - p\left(1 - (\langle\psi|\boldsymbol{W}|\psi\rangle)^2\right) \;\geq\; 1-p\ . \tag{7.64}$$

If a unitary error of the form eq. (7.63) 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. (7.45), then the state $|{\rm BAD}\rangle$ (which arises from terms in which $\boldsymbol{W}$ acts on at least $t+1$ qubits) has a norm of order $(\sqrt{p})^{t+1}$, and eq. (7.58) becomes

$$1 - F = O(p^{t+1})\ . \tag{7.65}$$

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 an angle θ 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 uncorrelated errors. In contrast, a unitary error that would cause more trouble is one of the form $\boldsymbol{U}^{(n)} \sim \boldsymbol{1} + i\theta \boldsymbol{E}^{(n)}_{\text{bad}}$, where $\boldsymbol{E}^{(n)}_{\text{bad}}$ is an n-qubit Pauli operator whose weight is greater than t. Then $|\text{BAD}\rangle$ has a norm of order θ, and the typical fidelity after recovery will be $F = 1 - O(\theta^2)$.

7.5 Classical Linear Codes

Quantum error-correcting codes were first invented less than four years ago, but classical error-correcting codes have a much longer history. Over the past fifty years, 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 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:

$$v(\alpha_1, \ldots, \alpha_k) = \sum_i \alpha_i v_i \ , \tag{7.66}$$

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

$$G = \begin{pmatrix} v_1 \\ \vdots \\ v_k \end{pmatrix} , \tag{7.67}$$

called the *generator matrix* of the code. Then in matrix notation, eq. (7.66) can be rewritten as

$$v(\alpha) = \alpha G \ ; \tag{7.68}$$

the matrix G, acting to the left, encodes the message α.

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

$$Hv = 0 \tag{7.69}$$

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

$$HG^T = 0 \ ; \tag{7.70}$$

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

$$v \to v + e \ . \tag{7.71}$$

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

$$H(v+e) = Hv + He = He \ . \tag{7.72}$$

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

$$v + e \to (v + e) + e = v \ . \tag{7.73}$$

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

$$v + e_1 \to v + (e_1 + e_2) \neq v. \tag{7.74}$$

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

$$0 = He_1 + He_2 = H(e_1 + e_2) \ , \tag{7.75}$$

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:

$$\sum_{v\in C}(-1)^{v\cdot u} = \begin{cases} 2^k & u \in C^\perp \\ 0 & u \notin C^\perp \end{cases}. \tag{7.76}$$

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

$$\sum_{v\in\{0,1\}^k}(-1)^{v\cdot w} = 0, w \neq 0, \tag{7.77}$$

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

$$v = \alpha G, \tag{7.78}$$

where α is a k-vector. Then

$$\sum_{v\in C}(-1)^{v\cdot u} = \sum_{\alpha\in\{0,1\}^k}(-1)^{\alpha\cdot Gu} = 0, \tag{7.79}$$

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 $u \notin C^\perp$.

7.6 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

$$|\bar{w}\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{v \in C_2} |v + w\rangle \ , \tag{7.80}$$

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 $\boldsymbol{H}^{(n)}$:

$$\begin{aligned} \boldsymbol{H}^{(n)} : \quad |\bar{w}\rangle_F &\equiv \frac{1}{\sqrt{2^{k_2}}} \sum_{v \in C_2} |v + w\rangle \\ \to \quad |\bar{w}\rangle_P &\equiv \frac{1}{\sqrt{2^{n}}} \sum_{u} \frac{1}{\sqrt{2^{k_2}}} \sum_{v \in C_2} (-1)^{u\cdot v} (-1)^{u \cdot w} |u\rangle \\ &= \frac{1}{\sqrt{2^{n-k_2}}} \sum_{u \in C_2^{\perp}} (-1)^{u\cdot w} |u\rangle \ ; \end{aligned} \tag{7.81}$$

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. (7.76)). 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

$$\begin{aligned} d_1 &\geq 2t_F + 1 \ , \\ d_2^{\perp} &\geq 2t_P + 1 \ . \end{aligned} \tag{7.82}$$

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

$$\boldsymbol{E}_e^{(\mathrm{flip})} : |v\rangle \to |v + e\rangle \ . \tag{7.83}$$

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

$$\boldsymbol{E}_e^{(\text{phase})} : |v\rangle \to (-1)^{v.e}|v\rangle \ , \tag{7.84}$$

which in the Hadamard rotated basis becomes

$$\boldsymbol{E}_e^{(\text{phase})} : |u\rangle \to |u+e\rangle \ . \tag{7.85}$$

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

$$|v\rangle \otimes |0\rangle_A \to |v\rangle \otimes |H_1 v\rangle_A \ , \tag{7.86}$$

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

$$|v\rangle \otimes |0\rangle_A \to |v\rangle \otimes |G_2 v\rangle_A \ , \tag{7.87}$$

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

$$\langle\bar{w}|\boldsymbol{E}_{e_P}^{(\text{phase})}\boldsymbol{E}_{e_F}^{(\text{flip})}|\bar{w}'\rangle = 0 \tag{7.88}$$

(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

$$d \;\geq\; \min(d_1, d_2^\perp)\ . \tag{7.89}$$

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. (7.86) (or eq. (7.87)) 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 ath bit of the syndrome, we prepare an ancilla bit in the state $|0\rangle_{A,a}$, and for each value of λ with $(H_1)_{a\lambda} = 1$, we execute a controlled-NOT gate with the ancilla bit as the target and qubit λ 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.

7.7 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 3×7

parity check matrix

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} . \tag{7.90}$$

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

$$H(e_1 + e_2) = He_1 + He_2 \neq 0, \tag{7.91}$$

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

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} ; \tag{7.92}$$

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

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

$$|\bar{1}\rangle_F = \frac{1}{\sqrt{8}} \sum_{\substack{\text{odd } v \\ \in \text{ Hamming}}} |v\rangle \ . \tag{7.93}$$

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

$$\begin{aligned} \boldsymbol{H}^{(7)} : |\bar{0}\rangle_F \to |\bar{0}\rangle_P &\equiv \left(\frac{1}{4}\right) \sum_{v \in \text{ Hamming}} |v\rangle = \frac{1}{\sqrt{2}}(|\bar{0}\rangle_F + |\bar{1}\rangle_F) \\ |\bar{1}\rangle_F \to |\bar{1}\rangle_P &\equiv \left(\frac{1}{4}\right) \sum_{v \in \text{ Hamming}} (-1)^{\text{wt}(v)}|v\rangle = \frac{1}{\sqrt{2}}(|\bar{0}\rangle_F - |\bar{1}\rangle_F). \end{aligned} \tag{7.94}$$

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

$$H(e_1 + e_2 + e_3) = 0 \ ; \tag{7.95}$$

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

$$|\bar{0}\rangle_F \leftrightarrow |\bar{1}\rangle_F. \tag{7.96}$$

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

$$|\bar{0}\rangle_P \leftrightarrow |\bar{1}\rangle_P, \tag{7.97}$$

or equivalently

$$\begin{aligned} |\bar{0}\rangle_F &\to |\bar{0}\rangle_F \\ |\bar{1}\rangle_F &\to -|\bar{1}\rangle_F, \end{aligned} \tag{7.98}$$

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.

7.8 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?

7.8.1 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

$$N(t) = \sum_{j=0}^{t} 3^j \binom{n}{j}. \tag{7.99}$$

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

$$N(t) = \sum_{j=0}^{t} 3^j \binom{n}{j} \leq 2^{n-k}. \tag{7.100}$$

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

$$1 + 3n \leq 2^{n-1}, \tag{7.101}$$

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.

7.8.2 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

$$n > 2(d-1)\ . \tag{7.102}$$

7.8.3 The quantum Singleton bound

We will now see that this result eq. (7.102) can be strengthened to

$$n - k \geq 2(d-1). \tag{7.103}$$

Eq. (7.103) resembles the Singleton bound on classical code parameters,

$$n - k \geq d - 1, \tag{7.104}$$

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.

An elegant proof of the quantum Singleton bound can be found that exploits the subadditivity of the Von Neumann entropy discussed in §5.2. We begin by introducing a k-qubit ancilla, and constructing a pure state that maximally entangles the ancilla with the 2^k codewords of the QECC:

$$|\Psi\rangle_{AQ} = \frac{1}{\sqrt{2^k}} \sum |x\rangle_A |\bar{x}\rangle_Q\ , \tag{7.105}$$

where $\{|x\rangle_A\}$ denotes an orthonormal basis for the 2^k-dimensional Hilbert space of the ancilla, and $\{|\bar{x}\rangle_Q\}$ denotes an orthonormal basis for the 2^k-dimensional code subspace. If we trace over the length-n code block Q, the density matrix ρ_A of the ancilla is $\frac{1}{2^k}\mathbf{1}$, which has entropy

$$S(A) = k = S(Q). \tag{7.106}$$

Now, if the code has distance d, then $d-1$ located errors can be corrected; or, as we have seen, no observable acting on $d-1$ of the n qubits can reveal any information about the encoded state. Equivalently, the observable can reveal nothing about the state of the ancilla in the entangled state $|\Psi\rangle$.

Now, since we already know that $n > 2(d-1)$ (if $k \geq 1$), let us imagine dividing the code block Q into three disjoint parts: a set of $d-1$ qubits $Q^{(1)}_{d-1}$, another disjoint set of $d-1$ qubits $Q^{(2)}_{d-1}$, and the remaining qubits $Q^{(3)}_{n-2(d-1)}$. If we trace out $Q^{(2)}$ and $Q^{(3)}$, the density matrix we obtain must contain no correlations between $Q^{(1)}$ and the ancilla A. This means that the entropy of system $AQ^{(1)}$ is additive:

$$S(Q^{(2)}Q^{(3)}) = S(AQ^{(1)}) = S(A) + S(Q^{(1)}). \tag{7.107}$$

Similarly,

$$S(Q^{(1)}Q^{(3)}) = S(AQ^{(2)}) = S(A) + S(Q^{(2)}). \tag{7.108}$$

Furthermore, in general, Von Neumann entropy is subadditive, so that

$$\begin{aligned} S(Q^{(1)}Q^{(3)}) &\leq S(Q^{(1)}) + S(Q^{(3)}) \\ S(Q^{(2)}Q^{(3)}) &\leq S(Q^{(2)}) + S(Q^{(3)}) \end{aligned} \tag{7.109}$$

Combining these inequalities with the equalities above, we find

$$\begin{aligned} S(A) + S(Q^{(2)}) &\leq S(Q^{(1)}) + S(Q^{(3)}) \\ S(A) + S(Q^{(1)}) &\leq S(Q^{(2)}) + S(Q^{(3)}). \end{aligned} \tag{7.110}$$

Both of these inequalities can be simultaneously satisfied only if

$$S(A) \leq S(Q^{(3)}) \tag{7.111}$$

Now $Q^{(3)}$ has dimension $n-2(d-1)$, and its entropy is bounded above by its dimension so that

$$S(A) = k \leq n - 2(d-1), \tag{7.112}$$

which is the quantum Singleton bound.

The $[[5,1,3]]$ code saturates this bound, but for most values of n and k the bound is not tight. Rains has obtained the stronger result that an $[[n,k,2t+1]]$ code with $k \geq 1$ must satisfy

$$t \leq \left[\frac{n+1}{6}\right], \tag{7.113}$$

(where $[x]$ = "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.

7.9 Stabilizer Codes

7.9.1 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

$$\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{X} = i\boldsymbol{\sigma}y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \tag{7.114}$$

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

$$\{\pm\boldsymbol{I}, \pm\boldsymbol{X}, \pm\boldsymbol{Y}, \pm\boldsymbol{Z}\} \equiv \pm\{\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}, \tag{7.115}$$

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

$$G_n = \pm\{\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}^{\oplus n}, \tag{7.116}$$

of order $|G_n| = 2^{2n+1}$ (since there are 4^n possible tensor products, and another factor of 2 for the ± 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

[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.

group G_n, and to its dimension-2^n faithful unitary representation by tensor products of 2×2 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:

(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: $\boldsymbol{MN} = \pm \boldsymbol{NM}$.

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,

$$|\psi\rangle \in \mathcal{H}_S \quad \text{iff} \quad \boldsymbol{M}|\psi\rangle = |\psi\rangle \text{ for all } \boldsymbol{M} \in S. \tag{7.117}$$

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 $\boldsymbol{M}$,

$$\boldsymbol{N}\boldsymbol{M} = -\boldsymbol{M}\boldsymbol{N} \; ; \tag{7.118}$$

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

$$\boldsymbol{M}_1|\psi\rangle = |\psi\rangle \; , \tag{7.119}$$

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

$$\boldsymbol{M}_1|\psi\rangle = \boldsymbol{M}_2|\psi\rangle = |\psi\rangle, \tag{7.120}$$

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

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

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

$$\boldsymbol{M}\boldsymbol{E}_a|\psi\rangle = -\boldsymbol{E}_a\boldsymbol{M}|\psi\rangle = -\boldsymbol{E}_a|\psi\rangle, \tag{7.122}$$

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

$$\boldsymbol{M}_i\boldsymbol{E}_a = (-1)^{s_{ia}}\boldsymbol{E}_a\boldsymbol{M}_i. \tag{7.123}$$

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

$$\langle\psi|\boldsymbol{E}_a^\dagger\boldsymbol{E}_b|\psi\rangle = C_{ab}, \tag{7.124}$$

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

$$\begin{aligned}\langle\psi|\boldsymbol{E}_a^\dagger\boldsymbol{E}_b|\psi\rangle &= \langle\psi|\boldsymbol{E}_a^\dagger\boldsymbol{E}_b\boldsymbol{M}|\psi\rangle \\ &= -\langle\psi|\boldsymbol{M}\boldsymbol{E}_a^\dagger\boldsymbol{E}_b|\psi\rangle = -\langle\psi|\boldsymbol{E}_a^\dagger\boldsymbol{E}_b|\psi\rangle,\end{aligned} \tag{7.125}$$

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.

7.9.2 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 ± sign) can be expressed as a product of $\boldsymbol{Z}$'s and $\boldsymbol{X}$'s; we may write

$$\boldsymbol{M} = \boldsymbol{Z}_M \cdot \boldsymbol{X}_M \tag{7.126}$$

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

$$(\alpha|\beta) \equiv \boldsymbol{Z}(\alpha)\boldsymbol{X}(\beta) = \bigotimes_{i=1}^{n} \boldsymbol{Z}^{\alpha_i} \cdot \bigotimes_{i=1}^{n} \boldsymbol{X}^{\beta_i}, \tag{7.127}$$

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

$$(\alpha|\beta)(\alpha'|\beta') = (-1)^{\alpha'\cdot\beta}(\alpha+\alpha'|\beta+\beta') \; ; \tag{7.128}$$

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. (7.127).

It follows from eq. (7.128) that the commutation properties of the Pauli operators can be expressed in the form

$$(\alpha|\beta)(\alpha'|\beta') = (-1)^{\alpha\cdot\beta'+\alpha'\cdot\beta}(\alpha'|\beta')(\alpha|\beta) \tag{7.129}$$

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

$$\alpha\cdot\beta' + \alpha'\cdot\beta \; . \tag{7.130}$$

We also note that the square of a Pauli operator is

$$(\alpha|\beta)^2 = (-1)^{\alpha\cdot\beta}\boldsymbol{I} \; , \tag{7.131}$$

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

$$\alpha\cdot\beta = 0\ . \tag{7.132}$$

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

$$\alpha\cdot\beta' + \alpha'\cdot\beta = (\alpha+\alpha')\cdot(\beta+\beta') - \alpha\cdot\beta - \alpha'\cdot\beta' = 0\ ; \tag{7.133}$$

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

$$H = (H_Z|H_X). \tag{7.134}$$

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

$$s_{ia} = (\alpha_a|\beta_a)\cdot(\alpha'_i|\beta'_i) = \alpha_a\cdot\beta'_i + \alpha'_i\cdot\beta_a. \tag{7.135}$$

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.

7.9.3 Some examples of stabilizer codes

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

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

$$\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3\boldsymbol{X}_4\boldsymbol{X}_5\boldsymbol{X}_6, \quad \boldsymbol{X}_4\boldsymbol{X}_5\boldsymbol{X}_6\boldsymbol{X}_7\boldsymbol{X}_8\boldsymbol{X}_9. \tag{7.136}$$

In the notation of eq. (7.134) these become

$$\left(\begin{array}{c|c|c|c}
\begin{matrix}1&1&0\\0&1&1\end{matrix} & 0 & 0 & \\
\cline{1-3}
0 & \begin{matrix}1&1&0\\0&1&1\end{matrix} & 0 & 0 \\
\cline{1-3}
0 & 0 & \begin{matrix}1&1&0\\0&1&1\end{matrix} & \\
\hline
\multicolumn{3}{c|}{0} & \begin{matrix}1&1&1&1&1&1&0&0&0\\0&0&0&1&1&1&1&1&1\end{matrix}
\end{array}\right)$$

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

$$\tilde{H} = \left(\begin{array}{c|c} H_{\rm ham} & 0 \\ 0 & H_{\rm ham}\end{array}\right), \tag{7.137}$$

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

$$\begin{aligned}
\boldsymbol{M}_1 &= \boldsymbol{Z}_1\boldsymbol{Z}_3\boldsymbol{Z}_5\boldsymbol{Z}_7\\
\boldsymbol{M}_2 &= \boldsymbol{Z}_2\boldsymbol{Z}_3\boldsymbol{Z}_6\boldsymbol{Z}_7\\
\boldsymbol{M}_3 &= \boldsymbol{Z}_4\boldsymbol{Z}_5\boldsymbol{Z}_6\boldsymbol{Z}_7,
\end{aligned} \tag{7.138}$$

detect the bit flips, and the three check operators

$$\begin{aligned} \boldsymbol{M}_4 &= \boldsymbol{X}_1\boldsymbol{X}_3\boldsymbol{X}_5\boldsymbol{X}_7 \\ \boldsymbol{M}_5 &= \boldsymbol{X}_2\boldsymbol{X}_3\boldsymbol{X}_6\boldsymbol{X}_7 \\ \boldsymbol{M}_6 &= \boldsymbol{X}_4\boldsymbol{X}_5\boldsymbol{X}_6\boldsymbol{X}_7, \end{aligned} \tag{7.139}$$

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_{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

$$\tilde{H} = \left(\begin{array}{c|c} H & 0 \\ 0 & H \end{array} \right) \tag{7.140}$$

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

$$|v\rangle \to |v+w\rangle, \tag{7.141}$$

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

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

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

$$H_X H_Z^T = H_Z H_X^T = 0 \ . \tag{7.143}$$

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

$$C_2 = C_X^\perp \subseteq C_1 = C_Z. \tag{7.144}$$

So we may characterize CSS codes as those and only those for which the stabilizer has generators of the form eq. (7.142).

However there is a caveat. The code defined by eq. (7.142) 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 $S^\perp \setminus S$.

7.9.4 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

$$\bar{\boldsymbol{Z}}_i \bar{\boldsymbol{X}}_j = (-1)^{\delta_{ij}} \bar{\boldsymbol{X}}_j \bar{\boldsymbol{Z}}_i. \tag{7.145}$$

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

$$\begin{aligned} \bar{\boldsymbol{Z}} &= \boldsymbol{X}_1 \boldsymbol{X}_2 \boldsymbol{X}_3 , \\ \bar{\boldsymbol{X}} &= \boldsymbol{Z}_1 \boldsymbol{Z}_4 \boldsymbol{Z}_7 . \end{aligned} \tag{7.146}$$

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

$$\begin{aligned} \bar{\boldsymbol{X}} &= \boldsymbol{X}_1 \boldsymbol{X}_2 \boldsymbol{X}_3 , \\ \bar{\boldsymbol{Z}} &= \boldsymbol{Z}_1 \boldsymbol{Z}_2 \boldsymbol{Z}_3 ; \end{aligned} \tag{7.147}$$

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. (7.93).

7.10 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

$$\begin{aligned}
\boldsymbol{M}_1 &= \boldsymbol{XZZXI}, \\
\boldsymbol{M}_2 &= \boldsymbol{IXZZX}, \\
\boldsymbol{M}_3 &= \boldsymbol{XIXZZ}, \\
\boldsymbol{M}_4 &= \boldsymbol{ZXIXZ},
\end{aligned} \tag{7.148}$$

$\boldsymbol{M}_{2,3,4}$ 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

$$\tilde{H} = \left(\begin{array}{c|c}
01100 & 10010 \\
00110 & 01001 \\
00011 & 10100 \\
10001 & 01010
\end{array}\right) \tag{7.149}$$

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 ith 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

$$\boldsymbol{M}_3\boldsymbol{M}_4 = -\boldsymbol{YXXYI}, \tag{7.150}$$

plus its cyclic permutations, and

$$\boldsymbol{M}_2\boldsymbol{M}_5 = -\boldsymbol{ZYYZI}, \tag{7.151}$$

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

For our logical operators, we may choose

$$\begin{aligned} \bar{\boldsymbol{Z}} &= \boldsymbol{ZZZZZ}, \\ \bar{\boldsymbol{X}} &= \boldsymbol{XXXXX}; \end{aligned} \tag{7.152}$$

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

$$\bar{\boldsymbol{Z}} = (\boldsymbol{ZZZZZ}) \cdot (-\boldsymbol{ZYYZI}) = -\boldsymbol{IXXIZ}, \tag{7.153}$$

(or one of its cyclic permutations), and

$$\bar{\boldsymbol{X}} = (\boldsymbol{XXXXX}) \cdot (-\boldsymbol{YXXYI}) = -\boldsymbol{ZIIZX}, \tag{7.154}$$

(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

$$|\Psi_0\rangle = \sum_{\boldsymbol{M} \in S} \boldsymbol{M}|\psi_0\rangle. \tag{7.155}$$

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)

$$\begin{aligned}
|\bar{0}\rangle &= \sum_{\boldsymbol{M}\in S} |00000\rangle \\
&= |00000\rangle + (\boldsymbol{M}_1 + \text{cyclic perms})\,|00000\rangle \\
&+ (\boldsymbol{M}_3\boldsymbol{M}_4 + \text{cyclic perms})\,|00000\rangle + (\boldsymbol{M}_2\boldsymbol{M}_5 + \text{cyclic perms})\,|00000\rangle \\
&= |00000\rangle + (110010\rangle + \text{ cyclic perms}) \\
&- (|11110\rangle + \text{ cyclic perms}) \\
&- (|01100\rangle + \text{ cyclic perms}). \qquad (7.156)
\end{aligned}$$

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

$$\begin{aligned}
|\bar{1}\rangle = \bar{\boldsymbol{X}}|\bar{0}\rangle &= |11111\rangle + (|01101\rangle + \text{ cyclic perms}) \\
&- (|00001\rangle + \text{ cyclic perms}) \\
&- (|10011\rangle + \text{ cyclic perms})\ . \qquad (7.157)
\end{aligned}$$

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

$$\boldsymbol{U}_{\text{encode}} : |0000\rangle \otimes (a|0\rangle + b|1\rangle) \to a|\bar{0}\rangle + b|\bar{1}\rangle. \qquad (7.158)$$

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)

$$a|\bar{0}\rangle + b|\bar{1}\rangle = \left(\sum_{\boldsymbol{M}\in S} \boldsymbol{M}\right) |0000\rangle \otimes (a|0\rangle + b|1\rangle). \qquad (7.159)$$

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

$$\sum_{\boldsymbol{M}\in S} \boldsymbol{M} = (\boldsymbol{I}+\boldsymbol{M}_4)(\boldsymbol{I}+\boldsymbol{M}_3)(\boldsymbol{I}+\boldsymbol{M}_2)(\boldsymbol{I}+\boldsymbol{M}_1)\ . \tag{7.160}$$

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 jth row to the ith row in the matrix $\tilde{H}$. With such row operations, we can perform a Gaussian elimination on the 4×5 matrix H_X, and so obtain the new presentation for the stabilizer

$$\tilde{H}' = \left(\begin{array}{c|c} 11011 & 10001 \\ 00110 & 01001 \\ 11000 & 00101 \\ 10111 & 00011 \end{array}\right) , \tag{7.161}$$

or

$$\begin{aligned} \boldsymbol{M}_1 &= \boldsymbol{YZIZY} \\ \boldsymbol{M}_2 &= \boldsymbol{IXZZX} \\ \boldsymbol{M}_3 &= \boldsymbol{ZZXIX} \\ \boldsymbol{M}_4 &= \boldsymbol{ZIZYY} \end{aligned} \tag{7.162}$$

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.

7.11 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 $m - 1$

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

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.)

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

$$\boldsymbol{\rho}^{(2)} = \frac{1}{4}\mathbf{1} \ . \tag{7.163}$$

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 $|\psi\rangle$

$$\langle\psi|\boldsymbol{E}|\psi\rangle \text{ independent of } |\psi\rangle, \quad \mathrm{wt}(\boldsymbol{E}) \leq n-m. \tag{7.164}$$

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

$$\langle\psi|\boldsymbol{E}|\psi\rangle \text{ independent of } |\psi\rangle, \quad \mathrm{wt}(\boldsymbol{E}) \leq m-1. \tag{7.165}$$

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

$$\begin{aligned} n-m<m \quad &\Rightarrow \quad n<2m\ , \\ m-1<n-m+1 \quad &\Rightarrow \quad n>2m-2\ . \end{aligned} \tag{7.166}$$

It follows that

$$n=2m-1\ , \tag{7.167}$$

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

$$[[2m-1,1,k]]_q \tag{7.168}$$

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.

7.12 Some Other Stabilizer Codes

7.12.1 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

$$|0\rangle \otimes |\bar{0}\rangle + |1\rangle \otimes |\bar{1}\rangle, \tag{7.169}$$

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.

7.12.2 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

$$\begin{array}{c} \boldsymbol{ZZ}\ , \\ \boldsymbol{XX}\ . \end{array} \tag{7.170}$$

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

$$\begin{array}{c} \boldsymbol{ZZZZ}\ , \\ \boldsymbol{XXXX}\ . \end{array} \tag{7.171}$$

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:

$$\begin{array}{ccl} |0000\rangle & + & |1111\rangle\ , \\ |0011\rangle & + & |1100\rangle\ , \\ |0101\rangle & + & |1010\rangle\ , \\ |0110\rangle & + & |1001\rangle\ . \end{array} \tag{7.172}$$

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

$$\begin{array}{ccc} \boldsymbol{Z}\boldsymbol{Z} & \ldots & \boldsymbol{Z}\ , \\ \boldsymbol{X}\boldsymbol{X} & \ldots & \boldsymbol{X}\ , \end{array} \tag{7.173}$$

(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

$$\frac{1}{\sqrt{2}}(|x\rangle + |\neg x\rangle), \tag{7.174}$$

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

7.12.3 The $[[8, 3, 3]]$ code

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

$$\tilde{H} = (H_Z|H_X), \tag{7.175}$$

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

$$\tilde{H} = \left(\begin{array}{c|c} H & H^\sigma \\ 11111111 & 00000000 \\ 00000000 & 11111111 \end{array} \right) . \tag{7.176}$$

Here H is the 3×8 matrix

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{7.177}$$

whose columns are all the distinct binary strings of length 3, and H^σ 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^σ are all distinct. We may see by inspection that a suitable choice is

$$H^\sigma = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \tag{7.178}$$

as the column sums are then

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

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^σ 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

$$H_{\text{EH}} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{7.180}$$

This matrix H_{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.

7.13 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:

$$1 + 3n = 2^{n-k}, \tag{7.181}$$

($16 = 16$ 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 $d = 3$,

$$1 + (q-1)n \leq q^{n-k}. \tag{7.182}$$

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

$$n = 2^m - 1, \;\; k = n - m, \tag{7.183}$$

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

$$n = \frac{q^m - 1}{q - 1} \;, \;\; k = n - m \;. \tag{7.184}$$

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

$$\begin{aligned} 1+1 &= \omega+\omega=\bar{\omega}+\bar{\omega}=0, \\ 1+\omega &= \bar{\omega}, \end{aligned} \tag{7.185}$$

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 $GF(4)^n$

$$(\alpha|\beta) \leftrightarrow \alpha + \beta\omega. \tag{7.186}$$

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

$$\alpha \cdot \beta' + \alpha' \cdot \beta\ . \tag{7.187}$$

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

$$v * u = \bar{v}_1 u_1 + \cdots + \bar{v}_n u_n\ , \tag{7.188}$$

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

$$v * u = a + b\omega \in GF(4)\ , \tag{7.189}$$

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

then our symplectic inner product is

$$v \cdot u = b \ . \tag{7.190}$$

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

$$v * (\bar{\omega}u) = b + a\bar{\omega} \tag{7.191}$$

so that

$$v \cdot (\bar{\omega}u) = a \ . \tag{7.192}$$

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

$$H = \begin{pmatrix} 1 & \omega & \omega & 1 & 0 \\ 0 & 1 & \omega & \omega & 1 \end{pmatrix}, \tag{7.193}$$

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 ω we obtain

$$\omega(1, \bar{\omega}, 0, \bar{\omega}, 1) = (\omega, 1, 0, 1, \omega), \tag{7.194}$$

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

$$\bar{\omega}(\omega, 0, \omega, \bar{\omega}, \bar{\omega}) = (1, 0, 1, \omega, \omega), \tag{7.195}$$

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

$$n = \frac{1}{3}(4^3 - 1) = 21. \tag{7.196}$$

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

$$[[21, 15, 3]], \tag{7.197}$$

that can correct one error.

7.14 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.

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

$$|\mathcal{E}^{(2)}| \cdot 2^{k-n}, \tag{7.198}$$

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

$$\log_2 |\mathcal{E}^{(2)}| \lesssim \log_2 \left[\binom{n}{2pn} 3^{2pn} \right] \sim n \left[H_2(2p) + 2p \log_2 3 \right]. \tag{7.199}$$

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

$$\log_2 |\mathcal{E}^{(2)}| + k - n < 0, \quad \text{or} \quad R < 1 - H_2(2p) - 2p \log_2 3. \tag{7.200}$$

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 $\le pn$.

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

7.15 Some Codes that Correct Multiple Errors

7.15.1 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

$$\boldsymbol{ZZI}\ , \quad \boldsymbol{IZZ}\ , \tag{7.201}$$

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

$$\boldsymbol{XXI}\ , \quad \boldsymbol{IXX} \tag{7.202}$$

(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

$$\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}\bar{\boldsymbol{I}},\ \bar{\boldsymbol{I}}\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}, \tag{7.203}$$

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

$$n = n_1 n_2 \ldots n_L, \tag{7.204}$$

and distance

$$d \geq d_1 d_2 \ldots d_L. \tag{7.205}$$

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

$$[[5^L, 1, 3^L]]. \tag{7.206}$$

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

$$\frac{t}{n} \sim \frac{1}{2}\left(\frac{3}{5}\right)^L , \tag{7.207}$$

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 §7.4.2, we can bound the failure probability by

$$p_{\text{fail}} \equiv p^{(1)} \leq \binom{5}{2} p^2 = 10p^2. \tag{7.208}$$

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. (7.208); we conclude that the probability of a botched recovery for the $[[25, 1, 9]]$ code is bounded above by

$$p^{(2)} \leq 10(p^{(1)})^2 \leq 10(10p^2)^2 = 1000p^4. \tag{7.209}$$

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

$$p^{(\ell)} \leq C[p^{(\ell-1)}]^2, \tag{7.210}$$

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

$$p^{(L)} \leq \frac{1}{C}(Cp)^{2^L}, \tag{7.211}$$

(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

$$p^{(L)} \leq p_o \left(\frac{p}{p_o}\right)^{2^L} , \tag{7.212}$$

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

$$p^{(L)} < \varepsilon, \tag{7.213}$$

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

$$n \leq \left[\frac{\log(p_o/\varepsilon)}{\log(p_o/p)}\right]^{\log_2 5} . \tag{7.214}$$

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.

7.15.2 Toric codes

The toric codes are another family of codes that, like concatenated codes, offer much better performance than would be expected on the basis of their distance. They'll be described by Professor Kitaev (who discovered them).

7.15.3 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,

$$f : \{0,1\}^m \to \{0,1\} \ . \tag{7.215}$$

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 the OR of two bits x and y can be expressed as

$$x \vee y == x + y - xy \ , \tag{7.216}$$

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

$$1,\ x_i,\ x_i x_j,\ x_i x_j x_k,\ \ldots\ , \tag{7.217}$$

(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 $x_{m-1}x_{m-2}\ldots x_1 x_0$

[4] See, *e.g.*, MacWilliams and Sloane, Chapter 13.

is $f(x_{m-1}, x_{m-2}, \ldots x_1, x_0)$. For example, for $m = 3$,

$$\begin{aligned} 1 &= (11111111) \\ x_0 &= (10101010) \\ x_1 &= (11001100) \\ x_2 &= (11110000) \\ x_0 x_1 &= (10001000) \\ x_0 x_2 &= (10100000) \\ x_1 x_2 &= (11000000) \\ x_0 x_1 x_2 &= (10000000)\ . \end{aligned} \tag{7.218}$$

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

$$k = 1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{r}. \tag{7.219}$$

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

$$f(x_m, \ldots, x_0) = g(x_{m-1}, \ldots, x_0) + x_m h(x_{m-1}, \ldots, x_0) \ , \tag{7.220}$$

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

$$f = (g|g) + (h|0) \tag{7.221}$$

where g, h are vectors of length 2^m. Consider the distance between f and

$$f' = (g'|g') + (h'|0) \ . \tag{7.222}$$

For $h = h'$ and $f \neq f'$ this distance is $\mathrm{wt}(f - f')$ $=2 \cdot \mathrm{wt}(g - g') \geq 2 \cdot$ dist $(R(r,m))$; for $h \neq h'$ it is at least $\mathrm{wt}(h - h') \geq$ dist $(R(r-1,m))$. If $d(r,m)$ denotes the distance of $R(r,m)$, then we see that

$$d(r, m+1) = \min\left(2\ d(r,m), d(r-1,m)\right) \ . \tag{7.223}$$

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

$$d(r, m+1) = \min(2^{m-r+1}, 2^{m-r+1}) = 2^{m-r+1}, \tag{7.224}$$

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

$$R(r,m)^\perp = R(m-r-1,m). \tag{7.225}$$

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 $r \le m-r-1$, or $2r \ge, m-1$, and self-dual for $2r = m-1$. In the self-dual case, the distance is

$$d = 2^{m-r} = 2^{\frac{1}{2}(m+1)} = \sqrt{2n}\ , \tag{7.226}$$

and the number of encoded bits is

$$k = \frac{1}{2}n = 2^{m-1}\ . \tag{7.227}$$

These self-dual codes, for $m = 3,5,7$, have parameters

$$[8,4,4], \quad [32,16,8], \quad [128,64,16]\ . \tag{7.228}$$

(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

$$[[8,0,4]], \quad [[32,0,8]], \quad [[128,0,16]]\ , \tag{7.229}$$

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

$$[[7,1,3]], \quad [[31,1,7]], \quad [[127,1,15]]\ , \tag{7.230}$$

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

$$\begin{aligned} n &= 2^m = 64 \\ d &= 2^{m-r} = 8 \\ k &= 1+6+\binom{6}{2}+\binom{6}{3} = 1+6+15+20 = 42 \ , \end{aligned} \tag{7.231}$$

is dual to $R(2,6)$, with parameters

$$\begin{aligned} n &= 2^m = 64 \\ d &= 2^{m-r} = 16 \\ k &= 1+6+\binom{6}{2} = 1+6+15 = 22 \ , \end{aligned} \tag{7.232}$$

and so the CSS construction yields a QECC with parameters

$$[[64,20,8]] \ . \tag{7.233}$$

Many other weakly self-dual codes are known and can likewise be employed.

7.15.4 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:

$$1 + \binom{23}{1} + \binom{23}{2} + \binom{23}{3} = 2^{11} = 2^{23-12}. \tag{7.234}$$

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)

$$0^1 8^{759} 12^{2576} 16^{759} 24^1 \ . \tag{7.235}$$

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

$$\binom{24}{5} \Big/ \binom{8}{5} = 759, \tag{7.236}$$

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

[5] MacWilliams and Sloane §6.10.

the sense that for any finite group G there is a unique decomposition of the form

$$G \equiv G_0 \supseteq G_1 \supseteq G_2 \geq \ldots \supseteq G_n, \tag{7.237}$$

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* Λ. 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 Λ, write each component x_j in binary notation,

$$x_j = \ldots x_{j3}x_{j2}x_{j1}x_{j0}\ . \tag{7.238}$$

Then $\vec{x} \in \Lambda$ if

(i) The x_{j0}'s are either all 0's or all 1's.

(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.*

$$\begin{aligned} -1 &= \ldots 11111\ , \\ -2 &= \ldots 11110\ , \\ -3 &= \ldots 11101\ , \\ &\quad\text{etc.} \end{aligned} \tag{7.239}$$

We can easily check that Λ 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:

$$\begin{aligned} (\pm 2)^8 &\ :\ 2^7 \cdot 759 \\ (\pm 3)(\mp 1)^{23} &\ :\ 2^{12} \cdot 24 \\ (\pm 4)^2 &\ :\ 2^2 \cdot \binom{24}{2}\ . \end{aligned} \tag{7.240}$$

That is, there are $759 \cdot 2^7$ neighbors that have eight components with the values ± 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 ± 3 (this component can be chosen in 24 ways) and the remaining 23 components have the value (∓ 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 ± 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

$$2^{22} \cdot 3^9 \cdot 5^4 \cdot 7^2 \cdot 11 \cdot 13 \cdot 23 = 8,315,553,613,086,720,000 \simeq 8.3 \times 10^{18}. \tag{7.241}$$

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 8.08×10^{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.

7.16 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

$$\mathcal{H}^{(n)} = \mathcal{H} \otimes \ldots \otimes \mathcal{H}. \tag{7.242}$$

We would like to select a code subspace $\mathcal{H}^{(n)}_{\text{code}}$ of $\mathcal{H}^{(n)}$ such that quantum information residing in $\mathcal{H}^{(n)}_{\text{code}}$ can be subjected to the superoperator

$$\$^{(n)} = \$ \otimes \ldots \otimes \$, \tag{7.243}$$

and yet can still be decoded with high fidelity.

The rate of a code is defined as

$$R = \frac{\log \mathcal{H}^{(n)}_{\text{code}}}{\log \mathcal{H}^{(n)}} \ ; \tag{7.244}$$

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 $\mathcal{H}^{(n)}_{\text{code}}$ with rate at least R, such that for any $|\psi\rangle \in \mathcal{H}^{(n)}_{\text{code}}$, the state $\boldsymbol{\rho}$ recovered after $|\psi\rangle$ passes through $\$^{(n)}$ has fidelity

$$F = \langle\psi|\boldsymbol{\rho}|\psi\rangle > 1 - \varepsilon. \tag{7.245}$$

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

$$\begin{aligned} |0\rangle \otimes |0\rangle_E &\rightarrow \sqrt{1-p}|0\rangle \otimes |0\rangle_E + \sqrt{p}|2\rangle \otimes |1\rangle_E, \\ |1\rangle \otimes |0\rangle_E &\rightarrow \sqrt{1-p}|1\rangle \otimes |0\rangle_E + \sqrt{p}|2\rangle \otimes |2\rangle_E, \end{aligned} \tag{7.246}$$

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.

7.16.1 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

$$r \leq m + Qn. \tag{7.247}$$

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

$$R = \frac{r}{m/Q + n} = \left(\frac{r}{m + Qn}\right) Q, \tag{7.248}$$

over the noisy channel. Were r to exceed $m + Qn$, this rate R would exceed the capacity, a contradiction. Therefore eq. (7.247) 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

$$Q(p_2) \leq 1 - \frac{p_2}{p_1} + \frac{p_2}{p_1} Q(p_1). \tag{7.249}$$

(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. (7.247) holds, and the rate R they can attain is bounded by

$$R \leq 1 - \frac{p_2}{p_1} + \frac{p_2}{p_1} Q(p_1). \tag{7.250}$$

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. (7.249) 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. (7.249). 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

$$Q(p) \leq 1 - 2p, \quad 0 \leq p \leq \frac{1}{2}. \tag{7.251}$$

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

$$P_{\text{fail}} \le 4^{pn}\left(\frac{1}{2}\right)^{n-k} = 2^{-n(1-2p-R)}. \tag{7.252}$$

where $R = k/n$ is the rate. Eq. (7.252) 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_{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. (7.251) we obtain the capacity of the quantum erasure channel:

$$Q(p) = 1-2p, \quad 0 \le p \le \frac{1}{2}\ . \tag{7.253}$$

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 §7.14 guarantees the existence of such a code for

$$R < 1 - H_2(p) - p\log_2 3. \tag{7.254}$$

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.

7.16.2 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 $p > 3/8$:

$$Q(p) = 0, \quad p > \frac{3}{8}. \tag{7.255}$$

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

$$|\psi\rangle\langle\psi| \to \left[(1-q)|\psi\rangle\langle\psi| + q\frac{1}{2}\mathbf{1}\right]^{\otimes 2}, \tag{7.256}$$

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 $p > 1/4$:

$$Q(p) = 0, \quad p > \frac{1}{4}. \tag{7.257}$$

Invoking the bound eq. (7.249) we infer that

$$Q(p) \le 1 - 4p, \quad 0 \le p \le \frac{1}{4}. \tag{7.258}$$

This result actually coincides with our bound on the rate of $[[n,k,d]]$ codes with $k \ge 1$ and $d \ge 2pn + 1$ found in §7.8. 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 $N_{\rm typ}$ of typical Pauli operators with total probability $1-\varepsilon$ can be bounded by

$$N_{\rm typ} \leq 2^{n(H_2(p)+p\log_2 3+\delta)}, \tag{7.259}$$

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

$$P_{\rm fail} \leq 2^{n(H_2(p)+p\log_2 3+\delta)} 2^{k-n} + \varepsilon \ . \tag{7.260}$$

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

$$R \equiv \frac{k}{n} \ < \ 1 - H_2(p) - p\log_2 3 - \delta'. \tag{7.261}$$

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

$$Q(p) \ \geq \ 1 - H_2(p) - p\log_2 3 \ . \tag{7.262}$$

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

$$\frac{n!}{(p_X n)!(p_Y n)!(p_Z n)!(p_I n)!} \sim 2^{nH(p_I,p_X,p_Y,p_Z)}, \tag{7.263}$$

where

$$H \equiv H(p_I, p_X, p_Y, p_Z) = -p_I \log_2 p_I - p_X \log_2 p_X - p_Y \log_2 p_Y - p_Z \log_2 p_Z, \tag{7.264}$$

is the Shannon entropy of the probability distribution $\{p_I, p_X, p_Y, p_Z\}$. Now we find

$$Q(p_I, p_X, p_Y, p_Z) \;\geq\; 1 - H(p_I, p_X, p_Y, p_Z)\;; \tag{7.265}$$

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.

7.16.3 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

[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.

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

$$\boldsymbol{Z}_1\boldsymbol{Z}_2,\ \boldsymbol{Z}_1\boldsymbol{Z}_3,\ \boldsymbol{Z}_1\boldsymbol{Z}_4,\ldots,\boldsymbol{Z}_1\boldsymbol{Z}_m. \tag{7.266}$$

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 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 ith qubit). Now the number of typical error operators acting on the n qubits is

$$2^{\sum_{i=1}^{n} H_i} \tag{7.267}$$

where

$$H_i = H(p_I^{(i)}, p_X^{(i)}, p_Y^{(i)}, p_Z^{(i)}), \tag{7.268}$$

is the Shannon entropy of the Pauli channel acting on the ith qubit. By the law of large numbers, we will have

$$\sum_{i=1}^{n} H_i = n\langle H\rangle, \tag{7.269}$$

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

$$R = \frac{1-\langle H\rangle}{m}, \tag{7.270}$$

(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. (7.262) 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 $[[5, 1, 3]]$ 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] 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).

7.17 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 $\mathcal{H}_{\rm code}$ 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 $|\psi\rangle\in\mathcal{H}_{\rm code}$ 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 $[[n_1n_2,1,d]]$ code with $d \geq d_1d_2$.

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.

7.18 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

$$\frac{1}{\sqrt{2}}\left(|0\rangle \otimes |\bar{0}\rangle + |1\rangle \otimes |\bar{1}\rangle\right) \ . \qquad (7.271)$$

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:

$$|\bar{0}\rangle = (|000\ldots 0\rangle + |111\ldots 1\rangle)^{\otimes m} \ ,$$
$$|\bar{1}\rangle = (|000\ldots 0\rangle - |111\ldots 1\rangle)^{\otimes m} \ . \qquad (7.272)$$

a) Construct stabilizer generators for this code, and construct the logical operations $\bar{\boldsymbol{Z}}$ and $\bar{\boldsymbol{X}}$ such that

$$\begin{aligned} \bar{\boldsymbol{Z}}|\bar{0}\rangle &= |\bar{0}\rangle \ , & \bar{\boldsymbol{X}}|\bar{0}\rangle &= |\bar{1}\rangle \ , \\ \bar{\boldsymbol{Z}}|\bar{1}\rangle &= -|\bar{1}\rangle \ , & \bar{\boldsymbol{X}}|\bar{1}\rangle &= |\bar{0}\rangle \ . \end{aligned} \tag{7.273}$$

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

$$\boldsymbol{U} : |\psi\rangle \otimes |0\rangle^{\otimes(n-k)} \rightarrow |\bar{\psi}\rangle \ , \tag{7.274}$$

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:

$$\begin{aligned} \boldsymbol{X}: &\quad |j\rangle \rightarrow |j+1 \ (\mathrm{mod}\ d)\rangle\ , \\ \boldsymbol{Z}: &\quad |j\rangle \rightarrow \omega^j |j\rangle\ , \end{aligned} \tag{7.275}$$

where $\omega = \exp(2\pi i/d)$. Then the $d \times d$ Pauli operators $\boldsymbol{E}_{r,s}$ are

$$\boldsymbol{E}_{r,s} \equiv \boldsymbol{X}^r \boldsymbol{Z}^s\ , \quad r,s = 0,1,\ldots,d-1 \tag{7.276}$$

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

$$\boldsymbol{E}_{r,s}\boldsymbol{E}_{t,u} = (\eta_{r,s;t,u})\boldsymbol{E}_{t,u}\boldsymbol{E}_{r,s}\ , \tag{7.277}$$

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

$$\text{SUM} : |j\rangle \otimes |k\rangle \rightarrow |j\rangle \otimes |k+j \pmod{d}\rangle \ . \tag{7.278}$$

a) Describe a quantum circuit containing SUM gates that can be executed to measure an n-qudit observable of the form

$$\bigotimes_a \boldsymbol{Z}_a^{s_a} \ . \tag{7.279}$$

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

$$\boldsymbol{U}_{r,s}\boldsymbol{E}_{r,s}\boldsymbol{U}_{r,s}^\dagger = \boldsymbol{Z} \ . \tag{7.280}$$

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

$$\bigotimes_a \boldsymbol{E}_{r_a,s_a} \ . \tag{7.281}$$

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

$$\boldsymbol{ZZZ} \ , \quad \boldsymbol{XXX} \ . \tag{7.282}$$

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

$$\begin{array}{ccccc} \boldsymbol{X} & \boldsymbol{Z} & \boldsymbol{Z}^{-1} & \boldsymbol{X}^{-1} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{X} & \boldsymbol{Z} & \boldsymbol{Z}^{-1} & \boldsymbol{X}^{-1} \\ \boldsymbol{X}^{-1} & \boldsymbol{I} & \boldsymbol{X} & \boldsymbol{Z} & \boldsymbol{Z}^{-1} \\ \boldsymbol{Z}^{-1} & \boldsymbol{X}^{-1} & \boldsymbol{I} & \boldsymbol{X} & \boldsymbol{Z} \end{array} \tag{7.283}$$

(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.)