A Study of Topological Quantum Error Correcting Codes
Part I: From Classical to Quantum ECCs

Preetum Nakkiran
preetum@berkeley.edu

Mar 23, 2015

Abstract

This survey aims to highlight some interesting ideas in Topological Quantum Error Correcting Codes, without assuming background in topology, quantum mechanics, or error-correcting codes.

Contents

1 Introduction

Here we only attempt to present one slice of a very large field, starting from scratch and building up to quantum surface codes. We will emphasize connections to classical coding theory along the way.

In Part I: We start by reviewing some classical coding theory, then introduce the quantum setting, and attempt to "lift" our understanding from the classical case. We will see several examples of quantum codes, and constructions similar to surface codes.

In Part II (planned): We will present surface codes (and possibly other topological quantum codes) and see how their topological, quantum, and coding-theoretic properties interact.

Some notable connections we will make along the way:

- Classical decoding by measuring syndrome $\longleftrightarrow$ Quantum decoding by measuring stabilizer.
- Stabilizer degenerate on codespace $\implies$ can be measured without collapsing codeword.
- Bit-flips $\longleftrightarrow$ classical code distance; phase-flips $\longleftrightarrow$ dual-code distance.
- Dual of surface code $\longleftrightarrow$ Code on dual lattice (Part II).

The material on QECCs here is mainly from the compilation [1], of which Chapters 2 and 19 (on QECCs in general, and topological QECCs) are available online from their authors, at [2] and [3]. The goal here is to present this material simply and directly, following only those aspects required to develop (and perhaps appreciate) quantum surface codes. Also, some of the elementary proofs have been expanded more explicitly, and the style is more by way of example (and motivated by the classical case).

Particularly important or interesting ideas are highlighted in the margins, like so.

2 Background

2.1 Classical ECCs

The following is a brief introduction to the basics of classical coding, including aspects which will be useful in the quantum case.

The goal of classical binary ECCs is to protect information from bit-flip errors. Traditionally, we imagine a sender and a receiver, who communicate across a noisy channel. That is, the channel will take input from the sender, flip some bits, and hand it to the receiver. We want to devise a communication scheme that is resilient to this noise.

More precisely, we wish to encode k bits (the *message*) into n bits (the *codeword*), such that we can always recover the original message from the codeword, even if it has been corrupted by at most t bit-flips. Error-correcting codes in general achieve this by embedding the message-space into the larger codespace such that the codewords are "far apart" – so the error patterns do not perturb codewords into each other.

We will consider linear codes (where encoding is a linear map $F_2^k \to F_2^n$) which are specified by a $n \times k$ *Generator matrix*. The simplest case is the $(3, 1)$-repetition code, which maps $k = 1$ bit to $n = 3$ bits as: $0 \mapsto 000$ and $1 \mapsto 111$. The generator matrix is simply:

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

The *minimum distance* of a code is the minimum Hamming-distance between any pair of codewords. For linear codes, this is equivalent to the codeword of minimum Hamming-weight. An $[n, k, d]$ code is one that maps k bits to n bits with minimum distance d. Codes within a given family are commonly specified as (n, k). The $(3, 1)$ repetition code has minimum distance 3. Clearly, this code can correct for 1 error.

In general, a code with minimum distance at least $2t + 1$ can correct for t errors: It not, there must be some codewords c_1, c_2 and errors e_1, e_2 (each with hamming-weight $w_H(e_i) \leq t$) such that $c_1 + e_1 = c_2 + e_2$ (that is, codeword c_1 with error e_1 is indistinguishable from codeword c_2 with error e_2). But then the hamming-distance $d_H(c_1, c_2) = w_H(c_1 - c_2) = w_H(e_1 - e_2) \leq 2t$, a contradiction to the minimum-distance.

We can also specify a linear code $\mathcal{C}$ in terms of its *parity-check matrix*, equal to the left nullspace of the generator matrix. That is, H is an $(n - k) \times n$ matrix s.t. $HG = 0$. H is thought of as specifying the constraints that codewords must satisfy, since

$$Hv = 0 \iff v \in \mathcal{C}$$

For the repetition code above,

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

The *dual code* of $\mathcal{C}$ (denoted $\mathcal{C}^\perp$) is the code with generator matrix equal to the pairity-check matrix of $\mathcal{C}$. Clearly $v \cdot w = 0 \; \forall v \in \mathcal{C}, w \in \mathcal{C}^\perp$.

Syndrome Decoding. The parity-check perspective leads to a decoding strategy known as *syndrome decoding*. Say the message m is sent as $y = Gm$, and received as $\tilde{y} = Gm + e$ for error-vector e. Hit the received vector with H, to find the *syndrome* $H\tilde{y} = H(Gm + e) = He$. Intuitively, multiplication by H collapses the codespace, and preserves only information about the error. And we can easily see that all correctable errors e can be identified by their syndrome He alone. For example, in the $(3, 1)$ repetition code, all 4 single-bit errors (including the identity) have distinct syndromes. Thus we can diagnose the error from the syndrome, then correct it to recover the original codeword (and hence, message). This idea will be crucial for QECCs, as we will see.

Hamming codes. Though we won't need them for developing QECCs, an example of a non-trivial classical ECC is the $(7, 4)$-Hamming code, with minimum distance 3:

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

2.2 Quantum Mechanics

The state of a quantum system is specified by a normalized vector in the Hilbert space L_2 (complex square-integrable functions, known as "wavefunctions"). Some notation: A state vector is denoted by the "ket" $|\psi\rangle$, and its dual (conjugate transpose, thought of as a row-vector), is denoted by the "bra" $\langle\psi| = |\psi\rangle^\dagger$. The inner-product of $|i\rangle$ and $|j\rangle$ is denoted $\langle i|j\rangle$.

For example, a two-level system has a 2-dimensional state-space. Let $|0\rangle$ and $|1\rangle$ be orthonormal basis vectors of this space. The system may be in states $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ where $\alpha, \beta \in \mathbb{C} : \langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1$. These coefficients α, β manifest physically when the system is measured.

Measurement. In QM, unlike classical physics, the state of a system is not directly reflected in some observable quantity. Rather, we may probe the state of a system by measuring "observables": the eigenvalues of Hermitian operators on the state. In QM, measurement is non-deterministic, and affects the system state, as in the following example: Say A is a Hermitian operator[1], with eigenvectors $|0\rangle, |1\rangle$ and eigenvalues $+1, -1$. Assume the system starts in a general state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Then when we measure A, we will get value $+1$ with probability $|\alpha|^2$ and -1 with probability $|\beta|^2$. Further, measurement *collapses* the state: If we measure $+1$, we will leave the system in the pure state of the corresponding eigenvector: $|0\rangle$. And if we measure -1, will leave it in $|1\rangle$. This is the nondeterministic part of QM.

Let us formalize measurement[2] to handle general observables. Consider the spectral decomposition of Hermitian operator A into projections P_i onto (orthogonal) eigenspaces with distinct eigenvalues λ_i:

$$A = \sum_i \lambda_i P_i$$

For a system in state $|\psi\rangle$, the probability of measuring value λ_i is $\rho_i = \langle\psi| P_i |\psi\rangle$. And after measurement of value λ_i, the system will collapse to state $P_i|\psi\rangle / \sqrt{\rho_i}$.

Notice that if no eigenvalue is degenerate, and we write $|\psi\rangle$ in the eigenbasis of A (call it $\{|i\rangle\}$), as: $|\psi\rangle = \sum_i \alpha_i |i\rangle$, then probability of measuring λ_i is just $|\alpha_i|^2$ (as in the two-level example before). This agrees with $\langle\psi|P_i|\psi\rangle$ since by orthogonality: $|\alpha_i|^2 = \alpha_i^* \langle i| \alpha_i |i\rangle = \left(\sum_j \alpha_j^* \langle j|\right) \alpha_i |i\rangle = \langle\psi| \alpha_i |i\rangle = \langle\psi|P_i|\psi\rangle$.

The key point is that measurement returns an eigenvalue of the observable operator, and collapses the system down to the corresponding eigenspace.

Evolution. The time-evolution of a system with wavefunction Ψ is given by the Schrödinger equation: $i\hbar\frac{\partial}{\partial t}\Psi = \hat{H}\Psi$. Here $\hat{H}$ is the Hamiltonion operator, which depends on the setup of the quantum system.

We will not use the details of this evolution here, except to note that the evolution is necessarily *linear* and *unitary*. That is, a system in state $|\psi\rangle$ can only evolve to states $U|\psi\rangle$, where U is a linear unitary operator: $U^\dagger U = I$. Unitarity can be derived from the Schrödinger equation, but also follows from the probabilistic interpretation of measurement (conservation of probability).

In quantum computation, the idea is to perform meaningful state-evolutions by setting up an appropriate Hamiltonion (eg, by modifying voltage potentials in a lab). We will assume we can realize various unitary transformations, which we use as "quantum gates."

Qubits. Finally, let us introduce the quantum system of choice in quantum computation. A *qubit* is two-level system – we will use the spin-1/2 particle. The basis states $|0\rangle, |1\rangle$ are eigenvectors of the Z Pauli spin matrix (the *computational basis*). The Pauli matrices are:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Together with I, these either commute or anti-commute with each other, and obey relations $Y = iXZ$ and so on. The eigenbasis of X is called the *plus-minus basis*: $|\pm\rangle = \frac{1}{\sqrt{2}}|0\rangle \pm \frac{1}{\sqrt{2}}|1\rangle$.

Notice X, Y, Z are all Hermitian – they correspond to observing the x, y, z components of the spin. Further, they are unitary, and so represent possible state-evolutions. Looking ahead, this will be our qubit error model. For example, $X|0\rangle = |1\rangle$, corresponding to a classical "bit flip" error.

We can combine many qubits into a joint system. If we consider qubits labeled A and B jointly, we get states in the tensor product basis: $|i\rangle_A \otimes |j\rangle_B$ for $i, j \in \{0, 1\}$. We write $|0\rangle_A \otimes |1\rangle_B$ as $|01\rangle$ and so on. We can

[1] A linear operator s.t. $A^\dagger = A$. Thus, all real eigenvalues and orthogonal eigenvectors.

[2] Called "Strong" or "Projective" measurements, because they give full information on the observable, and fully collapse the state. There are also "weak" measurements which give partial information...

tensor-product operators in the same way, eg: $X \otimes Z\,|01\rangle = (X\,|0\rangle) \otimes (Z\,|1\rangle) = |1\rangle \otimes (-\,|1\rangle) \equiv -\,|11\rangle$. It is possible to realize unitary operators which are NOT tensor products, by setting up Hamiltonians in which the qubits interact. This leads to *entangled states* (ie, states which are not tensor products), which is essential to QECCs and computation in general.

Pauli Group. The Pauli group $\mathcal{P}_n$ is represented by elements of the form $i^k\{I, X, Y, Z\}^{\otimes n}$; for example $\mathcal{P}_3$ contains $X \otimes I \otimes Z$. For all $P, Q \in \mathcal{P}_n$, the following hold:

1. $P^2 = \pm I$.
2. P, Q either commute or anti-commute.
3. $P^\dagger P = I$.

This will be a useful group for discussing operations on qubits.

An Example. Say we have prepared a qubit in state $|0\rangle$. If we measure Z (physically, the z-component of spin), we will always get $+1$, and the state will remain $|0\rangle$. In the X eigenbasis, this state is $|0\rangle = \frac{1}{\sqrt{2}}\,|+\rangle + \frac{1}{\sqrt{2}}\,|-\rangle$. So if we now measure X, we will get $+1$ w.p. 50%, and -1 w.p. 50% on this first measurement. Further, this will leave the qubit in pure state either $|+\rangle$ or $|-\rangle$. For example, if we measure X as -1, then all subsequent measurements of X will yield -1, since the state has collapsed to $|-\rangle$. If we then measure Z (from state $|-\rangle$), this will yield ± 1 w.p. 50% each, and leave the qubit in state either $|0\rangle$ or $|1\rangle$ accordingly.

3 Quantum Error Correction

3.1 Motivation

The main power of quantum computation comes from: instead of operating on only pure bits (0/1), it operates on *mixed states*, ie a superposition $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. (We will see examples of quantum circuits in developing QECCs).

Just as in the classical case, where we required an ECC protect bits from bit-flip errors, here we require a code to protect *qubits* from *quantum errors*. In particular, our code must protect arbitrary states: It is not sufficient to protect only pure states $|0\rangle$ and $|1\rangle$; we must be able to preserve the exact superposition $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ for each qubit. (After all, if our code only protected pure states, our quantum computer would be reduced in power to a classical computer).

3.2 Error Model

Analogous to classical bit-errors, we will consider *single-qubit errors*, which are just an X, Y, or Z operator on a single qubit. [3] X errors are called "bit-flips" and Z errors are called "phase-flips". For example, the error $Z \otimes I \otimes I$ is a phase-flip error on the first qubit, and takes $|100\rangle \mapsto -\,|100\rangle$.

3.3 First Attempts

Let us try to protect a single qubit from *bit-flip errors*, by encoding it with a repetition code (as we did in the classical case). We may try to simply "copy" the bit three times, enacting the transform $|\psi\rangle \mapsto |\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$. However, this transform is impossible, a result known as the *no-cloning theorem*.

[3] In fact, correcting X, Y, Z errors is sufficient to correct arbitrary errors on a single qubit. See Section 2.2.4 in [1]

Proof: We must have

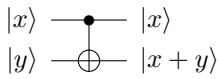
$$|0\rangle \mapsto |0\rangle^{\otimes 3}$$
$$|1\rangle \mapsto |1\rangle^{\otimes 3}$$

But then, by linearity:

$$\alpha|0\rangle + \beta|1\rangle \mapsto \alpha|0\rangle^{\otimes 3} + \beta|1\rangle^{\otimes 3} \neq (\alpha|0\rangle + \beta|1\rangle)^{\otimes 3}$$

□

Let us instead try encoding $|0\rangle \mapsto |000\rangle$, $|1\rangle \mapsto |111\rangle$. That is, instead of copying qubits, we encode them into a subspace. We will use the following CNOT gate:

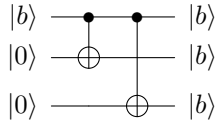


As illustrated, this transforms

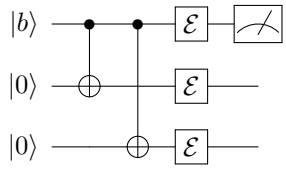
$$|x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes |x+y\rangle$$

(Addition $x + y$ in mod 2).

We can then encode by the following:



This takes $|0\rangle \mapsto |000\rangle$, $|1\rangle \mapsto |111\rangle$ as desired. These three qubits are then sent through the error-channel (denoted $\mathcal{E}$ below). How do we recover the original qubit at the end? In the classical case, we could measure all 3 qubits, and take the majority. This will not work here; consider what happens if we measure just the first qubit:



Assume no error occurs. Our codeword will be in the codespace spanned by $\{|000\rangle, |111\rangle\}$. Notice that measuring the first qubit is equivalent to measuring the eigenvalue of $Z \otimes I \otimes I$ (which will be -1 if the first qubit is $|1\rangle$, and $+1$ o.w.). We know this will collapse the state into an eigenspace: either $|0\cdot\cdot\rangle$ or $|1\cdot\cdot\rangle$. But this will destroy our encoded qubit: If we encoded $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \mapsto \alpha|000\rangle + \beta|111\rangle$, then measurement will collapse to the *pure state* $|000\rangle$ or $|111\rangle$. The problem here is, we cannot measure an operator that distinguishes between codewords, because it will collapse along the codespace. We can only measure something that does not does distinguish codewords, ie is degenerate on the codespace... this reminds us of pairity-checks in the classical case!

3.4 Stabilizers

The quantum analog of a pairity-check row (recall, a vector w s.t. $w \cdot v = 0\ \forall v \in \mathcal{C}$) is a *stabilizer*: An operator S s.t. $S|\psi\rangle = |\psi\rangle\ \ \forall |\psi\rangle \in \mathcal{C}$. That is, the $+1$ eigenspace of a stabilizer contains the codespace. As in the classical case, we can define our codespace as the intersection of such eigenspaces, by specifying a set of stabilizers.

Let us try to convert the pairity-checks of our classical repetition code into stabilizers of our quantum code. Recall the classical dual code was spanned by $\{\begin{bmatrix}1 & 1 & 0\end{bmatrix}, \begin{bmatrix}1 & 0 & 1\end{bmatrix}\}$. Thus, consider stabilizers $S_1 = Z \otimes Z \otimes I$ and $S_2 = Z \otimes I \otimes Z$. These generate the *stabilizer subgroup*, just as the classical pairity-check rows generated the nullspace. Now instead of measuring individual qubits, we will measure the stabilizers (as in syndrome decoding).

Again, say there are no errors, and we encode: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \mapsto \alpha|000\rangle + \beta|111\rangle$. Now when we measure each stabilizer S_i, we will get $+1$. And since the $+1$ eigenspace of each S_i contains the codespace, this measurement (projection) will preserve the codeword! Intuitively, the stabilizers do not distinguish between codewords, so measuring them will not collapse the codespace.

However, there is an additional subtlety here not present in the classical case. What happens when we have an error? Say we want to correct the single-qubit error $E = X \otimes I \otimes I$. This will take the input (sent) subspace $\mathcal{C} = \{|000\rangle, |111\rangle\}$ to output (received) subspace $E\mathcal{C} = \{|100\rangle, |011\rangle\}$. If we intend on decoding by measuring stabilizers, the subspace $E\mathcal{C}$ must be contained within eigenspaces of S_i. That is, the stabilizers should not distinguish between codewords *with errors* either (to prevent from collapsing the space). We got this for free in the classical case, by linearity: $w \cdot (v + e) = w \cdot v + w \cdot e = w \cdot e$ for all $v \in \mathcal{C}$ and error-vectors e. It is not true in general for stabilizers, but turns out to be true for single-qubit bit-flip errors in our example: [4]

Channel error	Received subspace	$S_1 = Z \otimes Z \otimes I$	$S_2 = Z \otimes I \otimes Z$
$I \otimes I \otimes I$	$\{\|000\rangle, \|111\rangle\}$	$+1$	$+1$
$X \otimes I \otimes I$	$\{\|100\rangle, \|011\rangle\}$	-1	-1
$I \otimes X \otimes I$	$\{\|010\rangle, \|101\rangle\}$	-1	$+1$
$I \otimes I \otimes X$	$\{\|001\rangle, \|110\rangle\}$	$+1$	-1

Notice that for each error pattern, the received subspace is still contained in an eigenspace of each S_i. Of course, just as in the classical case, we can diagnose the error from its measured syndrome (S_1, S_2).

In general, we have the following nice characterization of a *stabilizer quantum ECC*. Say we have a codespace $\mathcal{C}$ defined by a set of stabilizers $\{S_i\} \subset \mathcal{P}_n$ which commute with each other and generate the stabilizer group S. This code will be able to correct a set of unitary errors $\{E_i\} \subset \mathcal{P}_n$ if, for every pair of errors E_i, E_j, one of the following hold: [5]

1. $E_i^\dagger E_j \in S$.
2. $\exists M \in S : M E_i^\dagger E_j = -E_i^\dagger E_j M$. That is, M anti-commutes with $E_i^\dagger E_j$.
   (Equivalently, M commutes with exactly one of E_i, E_j).

The first condition means that E_i acts the same as E_j on the codespace:

$$E_i^\dagger E_j = M \in S \implies E_i^\dagger E_j |\psi\rangle = M|\psi\rangle = |\psi\rangle \ \ \forall |\psi\rangle \in \mathcal{C}$$

And

$$\begin{aligned} E_i^\dagger E_j |\psi\rangle &= |\psi\rangle \\ E_i E_i^\dagger E_j |\psi\rangle &= E_i |\psi\rangle \\ E_j |\psi\rangle &= E_i |\psi\rangle \qquad (E_i \text{ unitary}) \end{aligned}$$

[4] This example and table adapted from [2].

[5] This is the common phrasing of the theorem, since it relates to a more general characterization of QECCs. But the notation is unpacked below. See [4], Section 2.3 for more theoretical context. Finally, note that many of these properties hold because we restricted stabilizers and errors to be from the Pauli Group.

So in this case, we don't need to distinguish between E_i, E_j.

The content is in the second condition, which implies both the classical condition that different errors have different syndromes, and the quantum condition of not collapsing important subspaces. First, note that if operators M, T anti-commute, then T sends the $+1$ eigenspace of M to the -1 eigenspace:

$$M|\psi\rangle = |\psi\rangle \implies M(T|\psi\rangle) = -TM|\psi\rangle = -(T|\psi\rangle)$$

This tells us that measuring stabilizer S_i won't collapse the received codespace, even under an error E_j: Since both $E_j, S_i \in \mathcal{P}_n$, they either commute or anti-commute. If they commute, E_j sends the $+1$ eigenspace (codespace) of S_i back to the $+1$ eigenspace. If they anti-commute, we saw above that E_j sends the $+1$ eigenspace to the -1 eigenspace. Thus in either case, measuring S_i tells us nothing about the original codeword, even if error E_j occurred.

Further, if $M \in S$ anti-commutes with $E_i^\dagger E_j$, then measuring M lets us distinguish between error E_i and E_j occurring: Since E_i, E_j, M pairwise either anti-commute or commute, exactly one of $E_i^\dagger, E_j$ anti-commutes with M. WLOG, say $E_i^\dagger$ (and hence E_i) anti-commutes, but not E_j. Then

$$ME_i|\psi\rangle = -E_iM|\psi\rangle = -E_i|\psi\rangle$$

But

$$ME_j|\psi\rangle = E_jM|\psi\rangle = E_j|\psi\rangle$$

So measuring M will give -1 on error E_i, and $+1$ on error E_j, allowing us to distinguish between the two. Thus, we conclude that the two conditions are sufficient for a valid code, since distinguishable errors map to different syndromes, and we may measure syndromes without collapsing the received codespace (even under errors). We decode stabilizer codes by measuring syndromes, diagnosing the error pattern, and then applying an appropriate gate to invert the error.

Using this theory, we can more easily see that our repetition code works for bit-flip errors, by observing that the classical $(3, 1)$ linear repetition code corrects for 1 bit-flip error, so there is some syndrome[/stabilizer] distinguishing any two bit-flip errors in the classical[/quantum] case. This satisfies the second condition (distinguishing $E_i, E_j \iff$ commuting with exactly one of E_i, E_j). Similarly, any $[n, k, d]$ classical ECC can be used as an $[[n, k, d]]$ QECC that protects against (only) bit-errors. Notice that in the quantum case, codewords are all *superpositions* of classical codewords (in the computational basis).

3.5 Phase-flips

Our repetition code handles bit-flips, but what about phase-flips? Clearly the same code will not work, since the codeword $|000\rangle + |111\rangle$ (encoding of $|0\rangle + |1\rangle$) on error $Z \otimes I \otimes I$ yields $|000\rangle - |111\rangle$, which is another codeword (encoding of $|0\rangle - |1\rangle$). Fortunately, there is a nice trick that will convert a code for bit-flips into one for phase-flips.

Consider the "plus-minus" basis:

$$|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$$

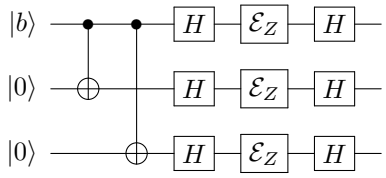
Notice that a phase-flip in the Z basis is a bit-flip in this basis, eg:

$$Z|+\rangle = \frac{1}{\sqrt{2}}(Z|0\rangle + Z|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$

Intuitively, we can protect our qubits from phase-flips by using a bit-flip code, but changing into the $|\pm\rangle$ basis before the channel. The *Hadamard gate* performs this change-of-basis: $|0\rangle, |1\rangle \xleftrightarrow{H} |+\rangle, |-\rangle$. The matrix representation is $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$; notice H is unitary and $H^2 = I$.

More formally: the Hadamard operator H has $HZH = X$. Thus the following two channels are equivalent:

Meaning, a X error in $\mathcal{E}_X$ causes an equivalent input-output behavior as a Z error in $\mathcal{E}_Z$. Therefore, we can convert a bit-flip code into a phase-flip code by applying H-gates on either side of the $\mathcal{E}_Z$ channel. For our repetition code, this looks like:



Notice this has the effect of transforming the codespace (sent across the channel) from $\{|000\rangle, |111\rangle\}$ to $\{|+++\rangle, |---\rangle\}$. More generally, the channel-equivalence tells us that we can analyze the effect of phase-flip errors on a code $\mathcal{C}$ by equivalently looking at bit-flip errors on the code $H^{\otimes n}\mathcal{C}$.

Aside: What about a code that can protect against both bit-flip and phase-flip errors? We will not detail it here, but one way to construct such a code is by *concatenating* our repetition codes: essentially using an inner-code to protect against bit-flips, and an outer-code to protect against phase-flips. This is the *9-qubit Shor Code* [6]. We will see more general constructions in the next section.

3.6 CSS Codes

We will now describe the Calderbank, Shor, Steane (CSS) code family, which constructs QECCs from suitable classical ECCs. Similar ideas will be used later in surface codes.

Intuition. First, consider what went wrong when we tried using our repetition code for phase-fips, as in intro of Section . Again, the problem was that the received state $|000\rangle - |111\rangle$ could be confused as either a true codeword, or an encoding of $|0\rangle + |1\rangle \mapsto |000\rangle + |111\rangle$ after error $Z \otimes I \otimes I$. Essentially, the problem was because we were encoding into individual basis elements, which could be individually flipped by errors (ie, our codespace was somewhat aligned with our error patterns). To fix this, what if we agreed to encode $|0\rangle$ as $|000\rangle + |111\rangle$? (Ignore the encoding of $|1\rangle$ for now). Then we will at least be able to detect phase-flips on this state, since if we received something like $|000\rangle - |111\rangle$, we'll know some error occurred (we agreed to send them both with the same sign). Of course, this won't work as a code, since we haven't specified how to encode $|1\rangle$. But this is the basic idea behind CSS codes: We take a classical code $\mathcal{C}$, and states become superpositions over *subcodes* of $\mathcal{C}$. (In our 1-dimentional repetition code, the only subcode was the entire code, so our code was trivial.)

Construction. As a first step, consider a state that is a superposition of codewords from an $[n, k, d]$ classical code $\mathcal{C}$:

$$|\psi\rangle = \frac{1}{\sqrt{2^k}} \sum_{v \in \mathcal{C}} |v\rangle$$

Now $|\psi\rangle$ will be protected from bit-flip errors, since it is a codeword in the QECC corresponding to $\mathcal{C}$ (recall our analysis of using classical ECCs as QECCs in Section 3.4).

To see what happens under phase-flips, we apply the Hadamard transform:

[6] See Section 2.2.5 in [1].

$$
\begin{aligned}
H^{\otimes n}|\psi\rangle &= H^{\otimes n}\frac{1}{\sqrt{2^k}}\sum_{v\in\mathcal{C}}|v\rangle \\
&= \frac{1}{\sqrt{2^k}}\sum_{v\in\mathcal{C}}H^{\otimes n}|v\rangle \\
&= \frac{1}{\sqrt{2^k}}\sum_{v\in\mathcal{C}}\bigotimes_{i\in[n]}H|v_i\rangle && \text{(unwrapping the tensor product)} \\
&= \frac{1}{\sqrt{2^k}}\sum_{v\in\mathcal{C}}\bigotimes_{i\in[n]}\frac{1}{\sqrt{2}}\Big(|0\rangle+(-1)^{v_i}|1\rangle\Big) && \text{(defn. of H)} \\
&= \frac{1}{\sqrt{2^k}}\sum_{v\in\mathcal{C}}\frac{1}{\sqrt{2^n}}\sum_{w\in F_2^n}(-1)^{v\cdot w}|w\rangle && \text{(dot-product mod 2)} \\
&= \frac{1}{\sqrt{2^{n+k}}}\sum_{w\in F_2^n}|w\rangle\sum_{v\in\mathcal{C}}(-1)^{v\cdot w}
\end{aligned}
$$

At this point, notice:

$$
\sum_{v\in\mathcal{C}}(-1)^{v\cdot w}=\begin{cases}2^k & \text{if} \quad w\in C^\perp\\ 0 & \text{o.w.}\end{cases}
$$

The $w\in C^\perp$ case is obvious; for the other case, note $\forall v_0\in\mathcal{C}: v_0+\mathcal{C}=\mathcal{C}$, since $\mathcal{C}$ is a linear subspace. And $w\notin C^\perp \implies \exists v_0\in C \quad \text{s.t.} \quad w\cdot v_0=1$. But then

$$
\sum_{v\in\mathcal{C}}(-1)^{v\cdot w}=\sum_{v\in\mathcal{C}}(-1)^{(v+v_0)\cdot w}=-\sum_{v\in\mathcal{C}}(-1)^{v\cdot w}
$$

So it must be 0. Therefore, we conclude:

$$
H^{\otimes n}|\psi\rangle=\frac{1}{\sqrt{2^{n-k}}}\sum_{w\in\mathcal{C}^\perp}|w\rangle
$$

Thus moving from bit-flips to phase-flips has taken us from a code to its dual! If $C^\perp$ has good distance, then $|\psi\rangle$ is protected from phase-flips, by the same reasoning as before (since the Hadamard transform let us view phase-flips as being bit-flips).

This was only to protect one state, but we can apply a similar construction to get a full code. Let $\mathcal{C}_1$ be an $[n,k_1,d_1]$ classical linear ECC that can correct t_1 errors. And let $\mathcal{C}_2\subset\mathcal{C}_1$ be an $[n,k_2,d_2]$ *subcode* ($k_2<k_1$) that can correct t_2 errors. The codewords of our QECC will be cosets of C_2 in C_1:

$$
|\overline{v}\rangle:=\frac{1}{\sqrt{2^{k_2}}}\sum_{w\in C_2}|w+v\rangle
$$

Here $|\overline{v}\rangle$ denotes the encoding of coset containing $v\in\mathcal{C}_1$, and the addition $w+v$ is mod 2. These codewords are orthonormal, since the cosets of $\mathcal{C}_2$ do not overlap [7]. Each codeword is a superposition of codewords in $\mathcal{C}_1$, so $|\overline{v}\rangle$ is protected from bit-flip errors. Further, applying Hadamard transform (similarly as above) shows:

$$
H^{\otimes n}|\overline{v}\rangle=\frac{1}{\sqrt{2^{n-k_2}}}\sum_{w\in\mathcal{C}_2^\perp}(-1)^{v\cdot w}|w\rangle
$$

[7] Warning: don't confuse orthogonality of $|v\rangle$ as vectors in *Hilbert-space* with orthogonality of v in F_2

Which is a superposition over $\mathcal{C}_2^\perp$. So $|\overline{v}\rangle$ is protected from phase-flip errors up to the distance of $\mathcal{C}_2^\perp$. Therefore, the QECC above is an $[[n, k_1 - k_2, \min(d_1, d_2)]]$ code, capable of correcting up to t_1 bit-flips and t_2 phase-flips. (This includes the case when bit-flips and phase-flips overlap, to make a $Y = iXZ$ error).

For concreteness, let us consider how decoding works. Notice that even after arbitrary phase-flips (by operator E), the codewords $E|\overline{v}\rangle \propto \sum_{w \in C_2} (\pm 1)|w + v\rangle$ are still superpositions of codewords $v \in \mathcal{C}_1$. Therefore, we can still correct bit-flip errors by measuring the syndromes of $\mathcal{C}_1$, and applying gates to invert the identified bit-flip error. That is, the addition of phase-flips does not affect our ability/procedure to correct for bit-flips. After correcting the bit-flips, we can again apply our usual phase-flip decoding: apply Hadamard transform, measure syndromes of $\mathcal{C}_2^\perp$ in the new basis, and diagnose/invert the error.

Intuition. For some intuition about why the dual code $\mathcal{C}^\perp$ is related to phase-flips, recall that by construction the phase-flip operators corresponding to vectors in the dual code are stabilizers of states in $\mathcal{C}$ (eg, $Z \otimes Z \otimes I$ stabilizes our repetition code). Thus the dual code consists of types of phase-flip errors which do not affect CSS codewords (superpositions of codewords in $\mathcal{C}$).

4 Preview: Surface Codes

As a preview of what's to come, the main ideas of surface codes are:

- Idea: Encode into the "homological degrees-of-freedom" of a meshed surface (eg, torus), with qubits on edges.
- Codewords are uniform superpositions over a homology class (similar to superpositions over a coset, in CSS codes).
- Stabilizers are appropriate "vertex" and "face" operators.
- Distance given by the shortest nontrivial cycle on the surface.
- The dual code is related to the dual lattice.

5 References

[1] D. A. Lidar and T. A. Brun, Eds., *Quantum Error Correction*, 1st ed. Cambridge University Press, 10 2013.

[2] D. Bacon. Introduction to quantum error correction. [Online]. Available: http://courses.cs.washington.edu/courses/cse599d/06wi/lecturenotes16.pdf

[3] H. Bombin, "An Introduction to Topological Quantum Codes," *ArXiv e-prints*, Nov. 2013. [Online]. Available: http://arxiv.org/abs/1311.0277

[4] D. Gottesman, "An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation," *ArXiv e-prints*, Apr. 2009. [Online]. Available: http://arxiv.org/pdf/0904.2557v1.pdf