Quantum Error Correction

Personal notes on quantum codes and fault tolerance

Himanshu Dongre

Based on the textbook by Daniel Gottesman

These are personal study notes based on Daniel Gottesman's *Surviving as a Quantum Computer in a Classical World*, May 7, 2024 draft, prepared as a supplement to UIC ECE 594: Quantum Error Correction, Fall 2025. They are not official textbook or course materials and are not endorsed by the author or institution. The selection, organization, annotations, and any errors are my own.

Definitions, theorem statements, selected arguments, and reproduced figures follow the textbook, with additional explanations and observations. These notes cover selected material through Chapter 15 of the 2024 draft; their section numbering differs from the textbook. No blanket publication license is asserted over third-party material.

Contents

I. Quantum Error Correcting Codes

1 Know Your Enemy

Quantum Errors

Definition. A *quantum channel* is a completely-positive trace-preserving (CPTP) map.

Definition. A channel of the form

$$\mathcal{R}_p(\rho) = (1-p)\rho + pZ\rho Z^\dagger$$

is a *dephasing channel*. When $p = \frac{1}{2}$, we have a *completely dephasing channel*. We usually restrict attention to $p \leq \frac{1}{2}$ since channels with $p > \frac{1}{2}$ are related to channels with $p < \frac{1}{2}$ by a Z operation.

Definition. The *depolarizing channel* is the quantum channel

$$\mathcal{D}_p(\rho) = (1-p)\rho + \frac{p}{3}X\rho X^\dagger + \frac{p}{3}Y\rho Y^\dagger + \frac{p}{3}Z\rho Z^\dagger\,.$$

$\mathcal{D}_{\frac{3}{4}}$ is the *completely depolarizing channel*.

Definition. A channel of the form

$$\mathcal{E}(\rho) = p_I\rho + p_X X\rho X^\dagger + p_Y Y\rho Y^\dagger + p_Z Z\rho Z^\dagger$$

is a *Pauli channel*.

Definition. An *amplitude damping channel* has the following form:

$$\mathcal{A}_p(\rho) = A_0\rho A_0^\dagger + A_1\rho A_1^\dagger\,,$$

where

$$A_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{bmatrix}.$$

Definition. A *qubit erasure error* is an error A acting on a single qubit which maps all qubit states to a third state $|\perp\rangle$ orthogonal to the qubit Hilbert space.

Definition. A channel of the form

$$\mathcal{E}(\rho) = \sum_P p_P P\rho P^\dagger\,,$$

where the sum is taken over P which are tensor products of I, X, Y, and Z, is a *multiple-qubit Pauli channel*.

Definition. An *independent* channel on n qudits (each of dimension q) has the form $\bigotimes_{i=1}^n \mathcal{E}_i$, where each $\mathcal{E}_i$ is a single-qudit channel.

Definition. We say that a linear operator A acting on n qubits has *weight* t if it is the tensor product of the identity on $n-t$ qubits with some matrix on the remaining t qubits. The weight of A is denoted $\mathrm{wt}\, A$. We say that a weight t operator A has *support* on the t qubits on which it acts non-trivially. Generally (but not always), we cite the weight and support for the minimal set of qubits for which A acts non-trivially. A linear operator B acting on n qubits is a *t-qubit error* if it has the form $B = \sum_i B_i$, where each B_i has weight at most t, not necessarily with support on the same set of qubits for different i. An n-qubit quantum channel is a *t-qubit error channel* if it has a Kraus decomposition for which all Kraus operators are t-qubit errors. We can similarly define a *t-qubit error map* acting linearly on n-qubit density matrices but which may not be trace perserving.

Definition. A *t-qubit erasure error* is a weight t operator which is the tensor product of erasure errors on the qubits in its support. A *t-qubit erasure channel* is a quantum channel with a Kraus representation where al Kraus operators are s-qubit erasure errors, with $s \leq t$. s can be different for different Kraus operators.

Theorem. Let $\mathcal{I}$ be the 1-qubit identity channel and $\mathcal{E} = \bigotimes_{i=1}^{n} \mathcal{E}_i$ be an n-qubit independent channel, with $\|\mathcal{E}_i - \mathcal{I}\|_\diamond < \varepsilon < \frac{t+1}{n-t-1}$, and $\varepsilon \leq \frac{1}{3}$ as well. Then

$$\|\mathcal{E} - \widetilde{\mathcal{E}}\|_\diamond < 5\binom{n}{t+1}[(4e+2)\varepsilon]^{t+1},$$

for some t-qubit error map $\widetilde{\mathcal{E}}$.

Proof sketch. *Admit.* (for now)

2 Redundancy Without Repetition

Basics Of Quantum Error Corecction

Theorem. *No-cloning.* There is no quantum operation which maps an arbitrary state $|\psi\rangle$ to $|\psi\rangle|\psi\rangle$.

Barriers to making a quantum error-correcting code:

1. No-cloning theorem prohibits repeating a quantum state.
2. Measuring the data while determining the error will destroy superpositions.
3. We must correct Y and Z errors in addition to X errors.
4. We must correct an infinite set of unitaries and also channels which decohere the state.

Definition. A *quantum error-correcting code (QECC)* $(\mathfrak{E}, \mathcal{E})$ is a partial isometry $\mathfrak{E} : \mathcal{H}_K \to \mathcal{H}_N$ with the set of *correctable errors* $\mathcal{E}$ (consisting of linear maps $E : \mathcal{H}_N \to \mathcal{H}_M$) with the following property: $\exists$ quantum operation $\mathfrak{D} : \mathcal{H}_M \to \mathcal{H}_K$ such that $\forall E \in \mathcal{E}, \forall |\psi\rangle \in \mathcal{H}_K$,

$$\mathfrak{D}(E\mathfrak{E}|\psi\rangle\langle\psi|\mathfrak{E}^\dagger E^\dagger) = c(E, |\psi\rangle)|\psi\rangle\langle\psi|\,.$$

$\mathfrak{E}$ is known as the *encoding* operation for the code (or the *encoder*), and $\mathfrak{D}$ is the *decoding* operation (or the *decoding map* or the *decoder*). We say that the QECC *corrects* an error

E if $E \in \mathcal{E}$. Sometimes, we don't consider a specific encoding map, and refer to Image($\mathfrak{E}$) (more precisely known as the *code space*) as the QECC. A *codeword* is any state in the code space. Often, a QECC is mentioned without an explicit set of correctable errors, which should be determined by context. Sometimes the error set simply does not matter. When it does matter but is not specified, the set of correctable errors is often the set of all t-qubit errors for the largets possible t; except sometimes it is instead the set of the most likely errors in the system. $\mathcal{H}_K$ is sometimes referred to as the *logical* Hilbert space and $\mathcal{H}_N$ is the *physical* Hilbert space. If $\mathcal{E}$ is the set of all t-qubit errors, we say that $\mathfrak{E}$ (or Image($\mathfrak{E}$)) is a *t-error correcting code*, or that it *corrects t errors*.

Proposition. In the definition of a QECC above, $c(E, |\psi\rangle)$ is independent of $|\psi\rangle$, as is $|A(E, |\psi\rangle)\rangle$ when we purify the decoder:

$$\mathscr{V}\left(E\left|\overline{\psi}\right\rangle_N \otimes |0\rangle_D\right) = \sqrt{c(E, |\psi\rangle)}|\psi\rangle_K \otimes |A(E, |\psi\rangle)\rangle_{D'} .$$

Proposition. Suppose we have two different encoders $\mathfrak{E}_1, \mathfrak{E}_2 : \mathcal{H}_K \to \mathcal{H}_N$ with Image($\mathfrak{E}_1$) = Image($\mathfrak{E}_2$). Then $(\mathfrak{E}_1, \mathcal{E})$ is a QECC if and only if $(\mathfrak{E}_2, \mathcal{E})$ is a QECC.

Theorem. If $(\mathfrak{E}, \mathcal{E})$ is a QECC, then $(\mathfrak{E}, \mathcal{E}')$ is a QECC, where $\mathcal{E}' = \operatorname{span} \mathcal{E}$. In other words, if a QECC can correct E and F, then it can also correct $\alpha E + \beta F$.

Corollary. If a QECC set of correctable errors includes all tensor products of I, X, Y, and Z of weight t or less, it is a t-error correcting code.

Theorem. Let $\mathcal{I}$ be the 1-qubit identity channel and $\mathcal{E} = \bigotimes_{i=1}^n \mathcal{E}_i$ be an n-qubit independent channel, with $\|\mathcal{E}_i - \mathcal{I}\|_\diamondsuit < \varepsilon < \frac{t+1}{n-t-1}$, and let $\mathfrak{E}$ and $\mathfrak{D}$ be the encoder and decoder for a QECC with n physical qubits that corrects t-qubit errors. Then

$$\|\mathfrak{D} \circ \mathcal{E} \circ \mathfrak{E} - \mathcal{I}\|_\diamondsuit < 2\binom{n}{t+1}(e\varepsilon)^{t+1} .$$

Proof sketch. *Admit.* (for now)

Suppose that $Q \subseteq \mathcal{H}_N$ is the code space, and let $E(Q)$ be the subspace formed by acting on Q with E. Then we want that for any pair of errors $E_1, E_2 \in \mathcal{E}$, the subspace $E_1(Q)$ is orthogonal to $E_2(Q)$. Then we will be able to make a measurement that identifies which subspace we are in and therefore identifies the error. Moreover, after identifying the error, we need to be able to reverse it. That is only possible if $E|_Q$ is a unitary map from Q to $E(Q)$. $E|_Q$ is unitary iff $\langle\psi|E^\dagger E|\phi\rangle = \langle\psi|\phi\rangle$ for all $|\psi\rangle, |\phi\rangle \in Q$. Thus, we get the sufficient condition that $(Q, \mathcal{E})$ is a QECC if $\forall |\psi\rangle, |\phi\rangle \in Q, \forall E_a, E_b \in \mathcal{E}$,

$$\langle\psi|E_a^\dagger E_b|\phi\rangle = \delta_{ab}\langle\psi|\phi\rangle .$$

A code that satisfies this condition is called a *nondegenerate (orthogonal) code*. However, this is not a necessary condition.

Theorem. *Knill-Laflamme Conditions.* $(Q, \mathcal{E})$ is a QECC iff $\forall |\psi\rangle, |\phi\rangle \in Q, \forall E_a, E_b \in \mathcal{E}$,

$$\langle\psi|E_a^\dagger E_b|\phi\rangle = C_{ab}\langle\psi|\phi\rangle .$$

Note that C_{ab} does not depend on $|\psi\rangle$ or $|\phi\rangle$.

Definition. Suppose $(\mathfrak{E}, \mathcal{E})$ is a QECC and $\mathcal{E}$ is linearly independent. Then the code is *degenerate* if $\text{rank}(C_{ab}) < |\mathcal{E}|$. A code $(\mathfrak{E}, \mathcal{E})$ (for linearly dependent $\mathcal{E}$) is degenerate if $(\mathfrak{E}, \mathcal{E}')$ is degenerate, where $\mathcal{E}'$ is a minimal spanning set for $\mathcal{E}$. If a code is not degenerate, it is *nondegenerate*.

$\star$ The essence of degeneracy is that different errors will produce the same result (or at least linearly dependent results) when acting on a codeword. For example, the Shor code is degenerate since any two Z errors acting on the same set of 3 qubits will produce the same result.

Definition. Let $Q \subseteq \mathcal{H}_N$ be a QECC. The *distance* of Q is the minimum weight d of an error F such that

$$\langle\psi|F|\phi\rangle \neq c(F)\langle\psi|\phi\rangle\,,$$

where $|\psi\rangle$ and $|\phi\rangle$ run over all possible pairs of codewords of Q.

Corollary. A distance d code corrects $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors.

$\star$ The converse of the corollary holds; i.e. a QECC that corrects t-qubit errors has distance $2t+1$.

Notation. A QECC which encodes $\mathcal{H}_K$ into $\mathcal{H}_{2^n}$ and has distance d is denoted as an $((n, K, d))$ code. If the physical Hilbert space is n qudits each of dimension q, it is an $((n, K, d))_q$ code. If the distance is unknown or irrelevant, it is an $((n, K))$ code or an $((n, K))_q$ code.

Definition. Let $Q \subseteq \mathcal{H}_N$ be a QECC with distance $d = 2t+1$. Then Q is *degenerate* if it is degenerate when the set of correctable errors is all t-qubit errors.

Definition. An encoder $\mathfrak{E}$ (defined as for a QECC) and a set of detectable errors $\mathcal{E}$ form a *quantum error-detecting code (QEDC)* if they have the following property: let Π be the projector on the code space. Then $\Pi E|\psi\rangle = c(E, |\psi\rangle)|\psi\rangle$, for all $E \in \mathcal{E}$ and all codewords $|\psi\rangle$. The code space is defined as Image($\mathfrak{E}$), as for a QECC, and we frequently refer to the code space as defining the error-detecting code instead of the encoder.

Theorem. *Knill-Laflamme Conditions.* $(\mathfrak{E}, \mathcal{E})$ is a QEDC if and only if

$$\langle\psi|E|\phi\rangle = c(E)\langle\psi|\phi\rangle$$

for all codewords $|\psi\rangle$ and $|\phi\rangle$ and all $E \in \mathcal{E}$. A code with distance d detects arbitrary $(d-1)$-qubit errors.

$\star$ A code able to correct t errors will also be able to detect $2t$ errors and vice-versa. Clearly QECCs and QEDCs aren't separate entities.

Definition. Given a subspace Q, its set of *detectable errors* is

$$\{E : \langle\psi|E|\phi\rangle = c(E)\langle\psi|\phi\rangle\ \forall|\psi\rangle, |\phi\rangle \in Q\}\,.$$

If $\mathcal{E}_C$ is the set of all correctable errors and $\mathcal{E}_D$ is the set of all detectable errors for a code, then the Knill-Laflamme conditions can be rephrased as $\mathcal{E}_C^2 \subseteq \mathcal{E}_D$.

Theorem. A QECC with distance d can correct $d-1$ erasure errors.

Proof.

1. Erasure errors are known-location errors unlike general errors where locations are unknown. In the erasure error model, the "which qubits are erased" information comes from a classical side channel—the environment or the channel itself tells you which qubits were affected without revealing anything about the encoded logical qubit. By definition of the erasure model, the locations of the erased qubits are given to the decoder as part of the channel output. The error syndrome (like positions of erasures) is classical information extracted by interacting with the environment.

2. If the erased qubits are in a set S, then the possible errors we might face are all operators supported only on S. Call that set of operators $\mathcal{E}_S$; i.e. $\mathcal{E}_S = \{E : \text{support}(E) \subseteq S\}$.

3. The KL conditions for correctability are $PE_a^\dagger E_b P = c_{ab} P$ for all E_a, E_b in the error set, where P is the projector onto the code space. Since in the erasure case, we know S, we only need these conditions to hold for errors whose support is within S; i.e. the KL conditions must hold for all $E_a, E_b \in \mathcal{E}_S$.

4. *Key fact:* $\mathcal{E}_S^2 = \mathcal{E}_S$; i.e. the set $\{E_i^\dagger E_j : E_i, E_j \in \mathcal{E}_S\}$ is the same as $\mathcal{E}_S$. This is because multiplying two operators supported on S still gives something supported on S.

5. From (3) and (4), we require that $PFP \propto P$ for all $F \in \mathcal{E}_S$. But these are precisely the KL conditions for detectability. These will hold iff $\text{wt}\, F = |S| < d$ for all $F \in S$. Thus, the code of distance d can correct erasures on any $t \leq d-1$ qubits.

Proposition. The following conditions are equivalent to the Knill-Laflamme conditions for correctability:

1) For all codewords $|\psi\rangle$, all pairs $E_a, E_b \in \mathcal{E}$,

$$\langle\psi|E_a^\dagger E_b|\psi\rangle = \text{tr}(|\psi\rangle\langle\psi|E_a^\dagger E_b) = C_{ab}\,.$$

2) If $\text{span}(\mathcal{E}) = \mathcal{E}$: For any pair of codewords $|\psi\rangle, |\phi\rangle$, any error $E \in \mathcal{E}$,

$$\langle\psi|E^\dagger E|\phi\rangle = C(E)\langle\psi|\phi\rangle\,.$$

3) If $\text{span}(\mathcal{E}) = \mathcal{E}$: For any codeword $|\psi\rangle$, any error $E \in \mathcal{E}$,

$$\text{tr}(|\psi\rangle\langle\psi|E^\dagger E) = C(E)\,.$$

4) Let $\{|i\rangle\}$ be a basis for the code space. For any i and j and for any pair $E_a, E_b \in \mathcal{E}$,

$$\langle i|E_a^\dagger E_b|j\rangle = C_{ab}\delta_{ij}\,.$$

⋆ Applying a similar argument to the definition of the set of detectable errors, we find that an operator E is detectable iff $\mathrm{tr}(\rho E)$ does not depend on the codeword ρ. This has an interesting interpretation: it says that an operator is detectable iff measurement of that operator reveals no information about the logical state of the code. Certainly, if measuring an operator does reveal information about the logical state, it will create errors in the encoded state. Two aspects of the condition are perhaps surprising: first, that some element of that error cannot be detected, and second, that revealing encoded information is the only thing that prevents an error from being detectable.

Proposition. Let $\mathcal{E}$ be a set of erasure errors, each one corresponding to a set of erased qubits (or qudits). Then Q corrects $\mathcal{E}$ iff ρ_S is the same for all logical states $|\psi\rangle$ whenever $S \in \mathcal{E}$. Here S is a subset of qubits that can be erased and ρ_S is the encoded state restricted to S. In other words, Q can correct for erasures on the set S iff the codeword on S has no information about the encoded state.

Definition. Two QECCs Q and Q' on n physical qubits are *equivalent* if Q' can be produced from Q by performing some set of single-qubit unitaries and by permuting the qubits. The notion of equivalence can of course be extended to codes on a q^n-dimensional Hilbert space as well.

Proposition. If Q and Q' are equivalent, then Q and Q' have the same number of logical qubits and the same distance.

3 Will The Real Codeword Please Stand Still?

Stabilizer Codes

Definition. The *Pauli group* P_n is composed of tensor products of I, X, Y, and Z on n qubits, with an overall phase of ± 1 or $\pm\iota$.

Definition. Let $P \in \mathsf{P}_n$. Then $\hat{P} \in \hat{\mathsf{P}}_n$ is the element of $\hat{\mathsf{P}}_n$ corresponding to P. Similarly, if S is a subset of P_n, then $\hat{S}$ is the subset of $\hat{\mathsf{P}}_n$ consisting of $\hat{P}$ for all $P \in S$.

Definition. Given a QECC $Q \subseteq \mathcal{H}_{2^n}$, the *stabilizer* $\mathsf{S}(Q)$ is

$$\mathsf{S}(Q) = \{M \in \mathsf{P}_n : M|\psi\rangle = |\psi\rangle \, \forall |\psi\rangle \in Q\}\,.$$

In other words, the stabilizer is composed of the Pauli operators for which all codewords are $+1$ eigenstates.

Proposition. The stabilizer $\mathsf{S}(Q)$ of a nonempty QECC Q has three basic properties:

1) $-I \notin \mathsf{S}(Q)$
2) $\mathsf{S}(Q)$ is a group
3) $\mathsf{S}(Q)$ is Abelian

Definition. Let $S \subseteq \mathsf{P}_n$ be an Abelian group, with $-I \notin S$. Then define the code space corresponding to S by

$$\mathsf{Q}(S) = \{|\psi\rangle : M|\psi\rangle = |\psi\rangle \, \forall M \in S\}\,.$$

In general, $Q \subseteq \mathsf{Q}(\mathsf{S}(Q))$.

Definition. If Q is a QECC with $Q = \mathsf{Q}(\mathsf{S}(Q))$, it is a *stabilizer code*. Stabilizer codes are also sometimes known as *symplectic codes*, *additive codes*, or GF(4) *additive codes*.

Proposition. $S = \mathsf{S}(\mathsf{Q}(S))$.

$\star$ Often, the stabilizer S is referred to as the code.

$\star$ For a stabilizer S with r generators, $|S| = 2^r$.

$\star$ The projector on the code space of S is $\Pi_S = \frac{1}{2^r}\prod_{i=1}^{r}(I + M_i) = \frac{1}{2^r}\sum_{M\in S} M$, where $\{M_1, \ldots, M_r\}$ is a generating set.

Proposition. If the stabilizer S on n physical qubits has $|S| = 2^r$ (i.e. S has r generators), then $\mathsf{Q}(S)$ has dimension 2^{n-r}; i.e. there are $k = n - r$ logical qubits.

Definition. A *stabilizer state* is the code space of a stabilizer with n generators on n qubits.

Definition. The *normalizer* $\mathsf{N}(S)$ of the stabilizer S is

$$\mathsf{N}(S) = \{N \in \mathsf{P}_n : NM = MN \,\forall M \in S\}.$$

In group theory, this is the definition of the *centralizer* of S (the set of things that commute with all elements of S) rather than the normalizer (the set of things that preserve S under conjugation), but because Paulis either commute or anticommute and $-I \notin S$, they are the same thing for a stabilizer.

Theorem. The set of undetectable errors for a stabilizer code S is $\hat{\mathsf{N}}(S) \setminus \hat{S}$. The distance of S is $\min_{\hat{E}\in\hat{\mathsf{N}}(S)\setminus\hat{S}} \operatorname{wt} E$.

Theorem. The stabilizer code S corrects a set of errors $\mathcal{E} \subseteq \mathsf{P}_n$ iff $\hat{E}^\dagger\hat{F} \notin \hat{\mathsf{N}}(S) \setminus \hat{S}$ for all $E, F \in \mathcal{E}$.

Notation. A stabilizer code with n physical qubits, k logical qubits, and distance d is denoted as an $[[n, k, d]]$ code. if the distance is unknown or irrelevant, it is an $[[n, k]]$ code.

Proposition. A stabilizer code S is degenerate for the error set $\mathcal{E}$ if and only if $\exists E_1, E_2 \in \mathcal{E}$ with $E_1^\dagger E_2 \in S$.

Proof sketch. $E_1^\dagger E_2 \in S \implies (E_1^\dagger F \in S \iff E_2^\dagger F \in S) \implies$ the rows corresponding to E_1 and E_2 in the matrix C in the KL conditions are identical.

Definition. A stabilizer code S with distance d is *degenerate* if $\exists M \in S,\ M \neq I$, with $\operatorname{wt} M < d$.

3.1 Cosets and Error Syndromes

Definition. Suppose S is a stabilizer code with generators $M_1, \ldots, M_r$ and $|\psi\rangle$ is an eigenvector (not necessarily with eigenvalue $+1$) of all $N \in S$. The *error syndrome* of $|\psi\rangle$ is an r-bit string with i^{th} bit 0 if $|\psi\rangle$ is a $+1$ eigenvector of M_i and i^{th} bit 1 if $|\psi\rangle$ is a -1 eigenvector of M_i. The error syndrome $\sigma(E) : \mathsf{P}_n \to \mathbb{Z}_2^r$ is the error syndrome of $E|\phi\rangle$ for

any codeword $|\phi\rangle$. When the stabilizer associated with a syndrome is in doubt, we will write $\sigma_S(E)$. If $P, Q \in \mathsf{P}_n$, then $s(P,Q) = 0$ if P and Q commute and $s(P,Q) = 1$ if they anticommute. ($s(P,Q) : \mathsf{P}_n \times \mathsf{P}_n \to \mathbb{Z}_2$). For a Pauli, bit i of the error syndrome is the same as $s(E, M_i)$.

Proposition. The $s(P,Q)$ and $\sigma(E)$ functions have the following properties. In all cases, $+$ refers to binary addition.

1) $s(P_1P_2, Q) = s(P_1, Q) + s(P_2, Q)$.
2) $s(P,Q) = s(Q,P)$.
3) $\sigma(EF) = \sigma(E) + \sigma(F)$.

Proposition. Let $E, F \in \mathsf{P}_n$, and S be a stabilizer. Then E and F are in the same coset of $\mathsf{N}(S)$ iff E and F have the same error syndrome.

Proposition. $|\mathsf{N}(S)| = 4 \cdot 2^{n+k}$ when S has n physical qubits and k logical qubits.

Proposition. Let $N_1, N_2 \in \mathsf{N}(S)$. Then N_1 and N_2 are in the same coset of S iff $N_1|\psi\rangle = N_2|\psi\rangle$ for all codewords $|\psi\rangle$.

Recall that when $N \in \mathsf{N}(S)$, then $N|\psi\rangle$ is a codeword for any input codeword $|\psi\rangle$. Thus, N is a *logical operation*: it always maps codewords to (possibly different) codewords.

Theorem. Let S be a stabilizer with n physical qubits and k logical qubits. Then $\mathsf{N}(S)/S \cong \mathsf{P}_k$. The quotient group $\mathsf{N}(S)/S$ can be taken to be the *logical Pauli group*.

Theorem. If S is a nondegenerate stabilizer code, the error syndromes of all errors in the correctable error set $\mathcal{E}$ are distinct. If S is a degenerate stabilizer code, E and F have the same error syndrome iff $\hat{E}^\dagger \hat{F} \in \hat{S}$.

Instead of having an error model where we fix a set $\mathcal{E}$ of errors that we'd like to correct, and don't settle for anything less than correcting all of them, we can have a model where error $E \in \mathsf{P}_n$ occurs with probability p_E in an n-qubit Pauli channel. Assume the latter for the following proposition.

Proposition. Let C_ς be the coset of $\hat{\mathsf{N}}(S)$ with error syndrome ς. $\hat{Q}_\varsigma \hat{S}$ is the coset of $\hat{S}$ containing the canonical error $\hat{Q}_\varsigma$ with syndrome ς, so $\hat{Q}_\varsigma \hat{S} \subseteq C_\varsigma$.

1) The probability of error syndrome ς (regardless of whether we correctly identify the error) is
$$p_\varsigma = \sum_{\hat{P} \in C_\varsigma} p_P \,.$$
2) The probability of having syndrome ς and no logical error (i.e. error correction is successful) is
$$p_{\varsigma,\checkmark} = \sum_{\hat{P} \in \hat{Q}_\varsigma \hat{S}} p_P \,.$$

3) The total probability of successfully decoding the state is

$$p_{\checkmark} = \sum_{\varsigma} p_{\varsigma,\checkmark} = \sum_{\varsigma} \sum_{\hat{P}\in\hat{Q}_\varsigma\hat{S}} p_P \,.$$

⋆ In order to maximize the probability of successfully decoding, for each syndrome ς, we should choose the coset $Q_\varsigma S$ which maximizes $p_{\varsigma,\checkmark}$. This maximum likelihood decoder should be contrasted with the decoder that optimizes the distance, for which we choose the coset containing the lowest-weight Pauli. If we have an independent Pauli channel, the two procedures frequently, but not always, give the same answer: When the probability of error per qubit is small, a specific low-weight error will have higher probability than a specific high-weight error. However, a coset which contains one low-weight error and a bunch of high-weight errors may be less likely than a coset which contains a large number of medium-weight errors.

3.2 Binary Symplectic Representation

Definition. The *binary symplectic representation (BSR)* of $\hat{\mathsf{P}}_n$ is an isomorphism between $\hat{\mathsf{P}}_n$ and $\mathbb{Z}_2^{2n} \times \mathbb{Z}_2^{2n} : P \leftrightarrow v_P = (x_P \| z_P)$. The i^{th} component of x_P is 0 if P acts on qubit i as I or Z and 1 if P acts on qubit i as X or Y. The i^{th} component of z_P is 0 if P acts on qubit i as I or X and 1 if P acts on qubit i as Y or Z. That is,

$$\begin{array}{cc} I & (0\|0) \\ X & (1\|0) \\ Y & (1\|1) \\ Z & (0\|1) \end{array} \quad \longleftrightarrow \quad .$$

For instance, $X \otimes I \otimes Y \leftrightarrow (1\ 0\ 1 \| 0\ 0\ 1)$. The "symplectic" refeers to the substitute for commutativity/anticommutativity.

Definition. The *symplectic form* (or symplectic product) on $\mathbb{Z}_2^{2n} \times \mathbb{Z}_2^{2n}$ is $(x_1\|z_1) \odot (x_2\|z_2) = x_1 z_2 + x_2 z_1$, where the product on the RHS is the usual scalar product in the vector space $\mathbb{Z}_2^n$ and addition is modulo 2.

Proposition. $v_p \odot v_Q = s(P,Q)$ for $P, Q \in \mathsf{P}_n$.

⋆ The one thing that is lost in the transition between represetations is the phase of a Pauli, so you must be careful whenever dealing with something that depends on that.

Definition. Let V be a linear subspace of $\mathbb{Z}_2^{2n} \times \mathbb{Z}_2^{2n}$. The *dual* of V (w.r.t. $\odot$) is $V^\perp = \{w \in \mathbb{Z}_2^{2n} \times \mathbb{Z}_2^{2n} : w \odot v = 0 \,\forall v \in V\}$. A subspace is *self-dual* if $V^\perp = V$. A subspace is *weakly self-dual* if $V \subseteq V^\perp$.

Definition. A set of Paulis $\{P_1, \ldots, P_m\}$ is *independent* iff the vectors $\{v_{P_1}, \ldots, v_{P_m}\}$ are linearly independent.

Lemma. Let $\{P_1, \ldots, P_m\}$ be an independent set of Paulis on n qubits, and let ϖ be an m-bit vector with components ϖ_i. Then $\exists Q \in \mathsf{P}_n$ such that $s(P_i, Q) = \varpi_i$. In fact, there are 2^{2n-m} such Paulis.

In the Pauli group	Binary symplectic representation
$P \in \mathsf{P}_n$	$v_P = (x_P \vert z_P) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^n$
Multiplication	Addition
$c(P, Q)$	$v_P \odot v_Q$
Phase	No equivalent
Stabilizer S	Weakly self-dual subspace S
Normalizer $\mathsf{N}(\mathsf{S})$	Dual subspace $\mathsf{S}^\perp$ (under $\odot$)
Minimal set of generators for S	Basis for S

Figure 1: Equivalence between the Pauli group and its BSR.

Proposition. Let S be a stabilizer with generators $\{M_i\}$. Then for any error syndrome ς, $\exists P \in \mathsf{P}_n$ with $\sigma(P) = \varsigma$. That is, not only is every Pauli outside $\mathsf{N}(S)$ associated with an error syndrome, but every possible error syndrome is associated with some Pauli.

4 Combining The Old And The New

Making Quantum Codes From Classical Codes

4.1 Calderbank-Shor-Steane Codes

Definition. A stabilizer code is a *Calderbank-Shor-Steane (CSS) code* if there is a choice of generators for which the stabilizer's BSR is of the form

$$\begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix},$$

where A is a $r_1 \times n$ matrix and B is a $r_2 \times n$ matrix for some r_1, r_2. The generators of the form $(b\|0)$ are X *generators* and the generators of the form $(0\|a)$ are Z *generators*. The CSS construction allows us to take two classical codes and make a quantum code.

$\star$ If we pick $A = B = H_3$, where H_3 is the parity check matrix for the $[7, 4, 3]$ Hamming code, we get a $[[7, 1, 3]]$ quantum code, called the *Steane code*.

Theorem. *Calderbank-Shor-Steane.* Let C_1 be an $[n, k_1, d_1]$ classical linear code with parity check matrix H_1 and C_2 be an $[n, k_2, d_2]$ classical linear code with parity check matrix H_2. Suppose $C_1^\perp \subseteq C_2$. Let S be the CSS code with stabilizer $\begin{bmatrix} 0 & H_1 \\ H_2 & 0 \end{bmatrix}$. Then S is an $[[n, k, d]]$ quantum code with $k = k_1 + k_2 - n$ and $d \geq \min(d_1, d_2)$.

4.2 GF(4) Codes

The one-qubit Pauli group has 4 elements, and so does the finite field GF(4). We can connect $\hat{\mathsf{P}}_n$ with an n-dimensional vector field over GF(4). For $n = 1$, we simply associate each

$+$	0	1	ω	ω^2
0	0	1	ω	ω^2
1	1	0	ω^2	ω
ω	ω	ω^2	0	1
ω^2	ω^2	ω	1	0

$\times$	0	1	ω	ω^2
0	0	0	0	0
1	0	1	ω	ω^2
ω	0	ω	ω^2	1
ω^2	0	ω^2	1	ω

Figure 2: The addition and multiplication tables for GF(4)

element of the Pauli group with the elements of GF(4):

$$\begin{array}{cc} I \leftrightarrow 0 & X \leftrightarrow 1 \\ Z \leftrightarrow \omega & Y \leftrightarrow \omega^2 \end{array}.$$

As with the BSR, we want some map from $\mathbb{F}_4 \times \mathbb{F}_4 \to \mathbb{Z}_2$ which gives 0 when one input is 0 or both inputs are the same, and gives 1 otherwise. It turns out that the correct formula is $\mathrm{tr}(\overline{a}b)$. The $\bar{\cdot}$ and trace operations are defined as follows for GF(4):

$$\begin{array}{cccc} \overline{0} = 0 & \overline{1} = 1 & \mathrm{tr}\, 0 = 0 & \mathrm{tr}\, 1 = 0 \\ \overline{\omega} = \omega^2 & \overline{\omega^2} = \omega & \mathrm{tr}\, \omega = 1 & \mathrm{tr}\, \omega^2 = 1 \end{array}.$$

The generalization to n-dimensional vectors is straightforward:

$$a * b = \mathrm{tr}(\overline{a} \cdot b)\,,$$

where $\cdot$ is just the usual dot product. In the classical coding literature, this inner product is sometimes known as the *trace-Hermitian inner product*.

In the Pauli group	In GF(4)
I	0
X	1
Z	ω
Y	ω^2
Multiplication	Addition
$c(P, Q)$	$a * b = \mathrm{tr}(\overline{a} \cdot b)$
Phase	No equivalent
No equivalent	Multiplication
Unitary T (see chapter 6)	Multiplication by ω
Stabilizer S	Weakly self-dual additive code S
Normalizer $\mathsf{N}(\mathsf{S})$	Dual $\mathsf{S}^{\perp}$ (under $*$)

Figure 3: Equivalence between the Pauli group and GF(4).

Theorem. Suppose C is an additive code over GF(4) which is weakly self-dual under $*$. Then C can be converted to a stabilizer S with parameters $[[n, k, d]]$, where n is the number

of physical registers in C. If C contains $K = 2^r$ codewords, then $k = n - r$. The distance of S is the smallest weight of a member of $C^\perp/C$, and in particular, d is at least equal to the distance of $C^\perp$.

Proposition. If C is a linear GF(4) code, then its dual w.r.t. the inner product $\overline{x} \cdot y$ is the same as the dual w.r.t. $*$.

Theorem. The duals (w.r.t. the standard inner product) of the Hamming codes over GF(4) can be converted to stabilizer codes. The dual of the $\left[\frac{4^r-1}{3}, \frac{4^r-1}{3} - r, 3\right]_4$ Hamming code becomes a $\left[\left[\frac{4^r-1}{3}, \frac{4^r-1}{3} - 2r, 3\right]\right]$ qubit stabilizer code.

Definition. A nondegenerate QECC is *perfect* if the number of correctable errors is exactly equal to the number of error syndromes.

E.g.:- The QECCs derived from the GF(4) Hamming codes are perfect qubit codes.

5 Symmetries Of Symmetries

The Clifford Group

Definition. The *Clifford group* C_n on n qubits is the normalizer of P_n in the unitary group $\mathsf{U}(2^n)$. That is,

$$\mathsf{C}_n = \{U \in \mathsf{U}(2^n) : UPU^\dagger \in \mathsf{P}_n \, \forall P \in \mathsf{P}_n\}\,.$$

The name "Clifford group" is motivated by the idea that there might be a connection of some sort to Clifford algebras, but the connection is not very close. To add to the confusion, you might encounter the term "Clifford group" in the mathematics literature referring to a different group. In quantumj information papers, Clifford group refers to C_n or one of its variants. You might also encounter the terms "normalizer group," "symplectic group," (or operations), or sometimes "stabilizer operations" for C_n. None of these terms is completely satisfactory, and "Clifford group" is the most widespread.

$\star$ By definition, the Pauli group P_n is a normal subgroup of C_n.

Definition. $\hat{\mathsf{C}}_n = \mathsf{C}_n/\{e^{\iota\theta} I\}$ (global phases ignored), $\check{\mathsf{C}}_n = \hat{\mathsf{C}}_n/\hat{\mathsf{P}}_n$ (Pauli group modded out).

$\star$ The best way of characterizing an element of the Clifford group is usually by giving the action under conjugation. Conjugation is a *group homomorphism*. Hence it is enough to specify the action under conjugation on a generating set of P_n. The action under conjugation on other elements can be deduced by taking products. Some examples of Clifford group elements:

1. Every Pauli $P \in \mathsf{P}_n \subset \mathsf{C}_n$ is a Clifford group element. It's effect on another Pauli $Q \in \mathsf{P}_n$ under conjugation is $PQP^\dagger = (-1)^{s(P,Q)}Q$.

2. The Hadamard transform H has the following conjugation action:

$$HXH = Z$$
$$HYH = -Y$$
$$HZH = X\,.$$

Note that $H^\dagger = H$. Also, the action of H on Y didn't need to be specified. It could've been deduced from the actions of H on X and Z.

3. Among two-qubit Clifford group elements, the most famous gate is the CNOT gate, which has the following conjugation acion on the Paulis:

$$X \otimes I \mapsto X \otimes X$$
$$Z \otimes I \mapsto Z \otimes I$$
$$I \otimes X \mapsto I \otimes X$$
$$I \otimes Z \mapsto Z \otimes Z\,.$$

This is a list of the action of CNOT on a generating set of four Paulis for P_2.

Theorem. Suppose U and V are unitaries which have the same action by conjugation on P_n: $UPU^\dagger = VPV^\dagger$ for all $P \in \mathsf{P}_n$. Then $U = e^{\iota\theta}V$ for some θ.

$\star$ Since conjugation is a group homomorphism, it preserves commutation and anticommutation; i.e. $s(UPU^\dagger, UQU^\dagger) = s(P, Q)$.

Procedure. Suppose the map $M : \mathsf{P}_n \to \mathsf{U}(2^n)$ is a group homomorphism, with $M(X_i) = \chi_i$ and $M(Z_i) = \zeta_i$ such that $s(\chi_i, \chi_j) = s(\zeta_i, \zeta_j) = 0$ and $s(\chi_i, \zeta_j) = \delta_{ij}$ (when χ_i and ζ_i are outside the Pauli group, the definition of $s(\cdot,\cdot)$ is the same, and $s(P, Q)$ is undefined here if P and Q neither commute nor anticommute). The following procedure finds the matrix representation in the standard basis of a U for which conjugation by U performs M:

1. Find the state $|\psi_0\rangle$ which is a $+1$ eigenstate of ζ_i for all $i = 1, \ldots, n$.
2. Let b be any number from 0 to $2^n - 1$, and let b_i be the i^{th} bit of b. Let $\chi(b) = \prod_i \chi_i^{b_i}$.
3. Let $|\psi_b\rangle = \chi(b)|\psi_0\rangle$.
4. Then $U_{ab} = \langle a|\psi_b\rangle$.

Proof sketch.

1. $Z_i = Z_i^\dagger \implies \zeta_i = \zeta_i^\dagger$ since M is a group homomorphism. Similarly, the χ_is are also Hermitian. Also, all of the ζ_is commute with each other, so $\Pi = \frac{1}{2^n}\prod_i (I + \zeta_i)$ is a projector with trace 1. Thus, there is a unique state $|\psi_0\rangle$ (up to global phase) which is a $+1$ eigenstate of all ζ_i as needed for step 1 of the procedure.
2. Show that the $|\psi_b\rangle$s form an orthonormal basis by showing that $b \neq b' \implies \langle\psi_b|\psi_b'\rangle = 0$ (use the definitions of $|\psi_b\rangle$, $|\psi_0\rangle$, and the commutation relations of ζ_j and χ_i).

3. A matrix element U_{ab} can be written as $\langle a|U|b\rangle$, where $|a\rangle$ and $|b\rangle$ are basis vectors in the standard computational basis. In the procedure, we define $U_{ab} = \langle a|\psi_b\rangle$, which conceptually means that U has the vectors $|\psi_b\rangle$s as its columns. This implies $\langle a|U|b\rangle = \langle a|\psi_b\rangle$ for all $a \implies U|b\rangle = |\psi_b\rangle$, and equivalently, $U^\dagger|\psi_b\rangle = |b\rangle$.

4. Hence, $UZ_iU^\dagger|\psi_b\rangle = UZ_i|b\rangle = (-1)^{b_i}U|b\rangle = (-1)^{b_i}|\psi_b\rangle = \zeta_i|\psi_b\rangle$. But the $|\psi_b\rangle$s form a basis $\implies UZ_iU^\dagger = \zeta_i$. Similarly, $UX_iU^\dagger = \chi_i$ can be verified to complete the proof that U is the (unitary) matrix representation in the standard basis, conjugation by which performs M.

In terms of the BSR, a group homomorphism on the Paulis becomes a linear map on $2n$-dimensional binary vectors such that the linear map has maximum rank (by the contraint that the generators map to independent Paulis) and the linear map preserves the symplectic inner product (by the constraint that conjugation preserves commutation relations). The symplectic inner product for binary vectors $v, w \in \mathbb{Z}_2^{2n}$ can be defined as $v \odot w = v^\mathsf{T} J w$, where $J = \begin{bmatrix} O & I_n \\ I_n & O \end{bmatrix}$. Hence, a linear map $M : \mathbb{Z}_2^{2n} \to \mathbb{Z}_2^{2n}$ preserves the symplectic inner product if $\forall v, w \in \mathbb{Z}_2^{2n}$,

$$\begin{aligned} & v \odot w = (Mv) \odot (Mw) \\ \implies & v^\mathsf{T} J w = v^\mathsf{T} M^\mathsf{T} J M w \,. \end{aligned}$$

Running over all v and w, this condition becomes $J = M^\mathsf{T} J M$. Define the *symplectic group* $\mathsf{Sp}(2n, \mathbb{Z}_2)$ as

$$\mathsf{Sp}(2n, \mathbb{Z}_2) = \{M \in \mathsf{GL}(2n, \mathbb{Z}_2) : J = M^\mathsf{T} J M\} \,,$$

where $\mathsf{GL}(2n, \mathbb{Z}_2)$ is the group of $2n \times 2n$ invertible binary matrices. Therefore, the punchline is that the linear map corresponding to the BSR of the conjugation map performed by a Clifford group operation is a member of the symplectic group $\mathsf{Sp}(2n, \mathbb{Z}_2)$. As usual, by going to the BSR, we lose all information about the phases of Paulis in the stabilizer. But, as it turns out, an operation which only changes the phases of Paulis is always performed by conjugation by a Pauli:

Proposition. If $UPU^\dagger = \pm P$ for all $P \in \mathsf{P}_n$, then $U = e^{\iota\theta} Q$ for $Q \in \mathsf{P}_n$ and some value of θ.

Theorem. $\check{\mathsf{C}} \cong \mathsf{Sp}(2n, \mathbb{Z}_2)$. Equivalently, for every map $X_i \mapsto \chi_i$, $Z_i \mapsto \zeta_i$, with $\chi_i, \zeta_i \in \mathsf{P}_n$ and satisfying the correct commutation relations, there exists $U \in \mathsf{C}_n$ which performs that map under conjugation, and U is unique up to overall phase.

5.1 Classical Simulation of the Clifford Group

One of the most interesting and useful things about the Clifford group is also one of the most disappointing. If $\in \mathsf{C}_n$, we can specify it by giving a global phase plus the images of $2n$ Paulis $X_1, Z_1, \ldots, X_n, Z_n$. Each image is also an n-qubit Pauli, so requires at most $2n + 1$ bits to specify. Thus, a Clifford group element can be specified using only $(2n + 1)(2n)$ bits plus a global phase. Contrast that with a general unitary $U \in \mathsf{U}(2^n)$, which needs 2^{2^n} real parameters to specify. Indeed, circuits made out of Clifford group gates have a much stronger

property: they can be efficiently simulated on a classical computer. This is a very useful fact, since it makes working with even uniformly large Clifford group circuits tractable. However, there is a price to be paid because Clifford group gates can classically be simulated, it means that a circuit composed of Clifford group gates cannot access the full power of quantum computation.

Procedure. Given a circuit consisting of a product $U = \prod_{i=m}^{1} U_i$, with $U_i \in \mathsf{C}_n$ (i.e. U_1 is the first gate performed, and U_m is the last gate to be performed), where each U_i is specified by its action on the generators of the Pauli group ($U_i : X_j \mapsto U_i(X_j), Z_j \mapsto U_i(Z_j)$), the action of the overall circuit U on the Pauli group can be determined as follows:

1. **let** $\chi_j \leftarrow X_j$
2. **let** $\zeta_j \leftarrow Z_j$
3. **for** $i = 1$ to m
    1. $\chi_j \leftarrow U_i(\chi_j)$
    2. $\zeta_j \leftarrow U_i(\zeta_j)$
4. **return** $X_j \mapsto \chi_j$, $Z_j \mapsto \zeta_j$ `// this is the action of the overall circuit` U

$\star$ Each gate can be updated in a total time $O(n)$ (since we need to update $2n$ χ and ζ operators). The simulation of the full circuit thus takes time $O(nm)$.

Procedure. We are given a circuit consisting of a product $U = \prod_{i=m}^{m} U_i$, with $U_i \in \mathsf{C}_i$, which is to be performed on an arbitrary input state from stabilizer code Q. Q encodes k qubits $0 \leq k \leq n$, has generators $M_1, \ldots, M_{n-k}$ and has logical Pauli operators $\overline{X}_j, \overline{Z}_j, j = 1, \ldots, k$. Each U_i can be specified by its action on the generators of the Pauli group ($U_i : X_j \mapsto U_i(X_j), Z_j \mapsto U_i(Z_j)$). Then the action of the overall circuit on the stabilizer code can be determined as follows:

1. **let** $\mu_j \leftarrow M_j$ $(j = 1, \ldots, n-k)$
2. **let** $\overline{\chi}_j \leftarrow \overline{X}_j$ $(j = 1, \ldots, k)$
3. **let** $\overline{\zeta}_j \leftarrow \overline{Z}_j$ $(j = 1, \ldots, k)$
4. **for** $i = 1$ to m
    1. **for** $j = 1$ to k
        1. $\overline{\chi}_j \leftarrow U_i(\overline{\chi}_j)$
        2. $\overline{\zeta}_j \leftarrow U_i(\overline{\zeta}_j)$
    2. **for** $j = 1$ to $n-k$
        1. $\mu_j \leftarrow U_i(\mu_j)$
    3. $S \leftarrow \langle \mu_1, \ldots, \mu_{n-k} \rangle$ `// current stabilizer`
    4. If desired, choose a new set of generators for S (update the μ_js)
    5. If desired, rewrite $\overline{\chi}_j \leftarrow \mu\overline{\chi}_j$ or $\overline{\zeta}_j \leftarrow \mu\overline{\zeta}_j$ with $\mu \in S$
5. **return** $S = \langle \mu_1, \ldots, \mu_{n-k} \rangle$, $(\overline{X}_j \mapsto \overline{\chi}_j, \overline{Z}_j \mapsto \overline{\zeta}_j)$

The output state lies in the stabilizer code $\mathsf{Q}(S)$. The encoded state has undergone the Clifford group operation given by the transformation $\overline{X}_j \mapsto \overline{\chi}_j$, $\overline{Z}_j \mapsto \overline{\zeta}_j$.

Proposition. Let S be a stabilizer, and let $P \notin \mathsf{N}(S)$ be a Pauli that does not commute with every element of S. Then exactly half of the elements of S commute with P. Furthermore, for any coset $\overline{Q} \in \mathsf{N}(S)/S$, exactly half the elements of $\overline{Q}$ commute with P.

Corollary. Let S be a stabilizer and P be a Pauli such that $P \notin \mathsf{N}(S)$. Define a stabilizer S_P generated by P plus all $N \in S$ such that $[N, P] = 0$. Then, $|S| = |S_P|$.

When we measure the eigenvalue of an arbitrary Pauli on n qubits, it is equivalent to the measurement of single qubits in the standard basis. This is because measurement of any Pauli can be implemented via single-qubit measurements plus Clifford group operations. For simplicity, we restrict attention to Paulis with overall phase ± 1, so that eigenvalues are ± 1. Suppose S is the current stabilizer (and the current state is a codeword in $\mathsf{Q}(S)$) then there are three cases to consider:

1. $\pm P \in S$: The measurement outcome is just ± 1 ($+1$ if $P \in S$ and -1 if $-P \in S$), and the state does not change, since the dtate being measured is an eigenstate of P.

2. $P \in \mathsf{N}(S)$: The measurement of P constitutes a measurement of some of the encoded data (since every $P \in \mathsf{N}(S)$ is a representative of a logical Pauli operation). We must rewrite P as a product of the logical Paulis $\overline{X}$ and $\overline{Z}$, which determines that P corresponds to some logical Pauli $\overline{Q}$. The measurement output distribution of P is the same as the measurement output distribution of Q on the original encoded state. We must also update the encoded state for the effects of the measurement. Finally, the measurement has outcome ± 1, so we know that the post-measurement state is a $+1$ eigenstate of $\pm P$. In other words, P has joined the stabilizer.

3. $P \notin \mathsf{N}(S)$: This implies $\exists M \in S$ such that $\{P, M\} = 0$. Suppose $|\psi\rangle \in \mathsf{Q}(S)$, then $M|\psi\rangle = |\psi\rangle$. The expectation value of a measurement of P on $|\psi\rangle$ is:

$$\begin{aligned}\langle\psi|P|\psi\rangle &= \langle\psi|PM|\psi\rangle \\ &= \langle\psi|(-PM)|\psi\rangle \\ &= -\langle\psi|P|\psi\rangle \\ &= 0\end{aligned}$$

   Thus, the probability of a $+1$ outcome and a -1 outcome afor the measurement are both $\frac{1}{2}$. Suppose the measurement has outcome $(-1)^b$, let $P' = (-1)^b P$. Then, the state afterwards is $|\psi'\rangle = \frac{1}{\sqrt{2}}(I + P')|\psi\rangle$ (taking into account normalization). If $M \in S$ commutes with P, then $|\psi'\rangle$ is still a $+1$ eigenstate of M. $|\psi'\rangle$ is also a $+1$ eigenstate of P'. That means that it is not an eigenstate of any M with $\{M, P\} = 0$, even if $M \in S$. The punchline is that the post-measurement stabilizer is $S_{P'}$. Furthermore, the S-coset of the logical $\overline{Z}_i$ operator gets replaced by the $S_{P'}$-coset which contains an element of the original $\overline{Z}_i$ that commutes with P. Similarly, we can find the new versions of the other logical Paulis. In short, we know how to update both the stabilizer and the logical Paulis after measurement of a Pauli operator.

Theorem. Suppose our system is a codeword of the stabilizer S with generators $M_1, \ldots, M_{n-k}$, and logical Paulis $\overline{X}_i$ and $\overline{Z}_i$ ($i = 1, \ldots, k$), and we measure the eigenvalue of Pauli $P \notin \mathsf{N}(S)$. Then, conditioned on outcome $(-1)^b$, the stabilizer and logical Paulis of the system can be determined as follows:

1. Find generator $N \in \{M_j\}$ such that $\{N, P\} = 0$. Reorder the generators so that $M_1 = N$.

2. For each generator M_i, $i > 1$, let $M_i' = M_i N^{s(M_i,P)} = \begin{cases} M_i & \text{if } [M_i, P] = 0 \\ M_i N & \text{if } \{M_i, P\} = 0 \end{cases}$.
3. Let $M_1' = (-1)^b P$.
4. Define the new stabilizer $S' = \langle M_1', \ldots, M_{n-k}' \rangle$.
5. Pick a representative $\overline{Z}_i$ for each of the logical Z operators for S. Let $\overline{Z}_i' = \overline{Z}_i M^{s(\overline{Z}_i,P)}$. $\overline{Z}_i'$ is a representative for the new logical Z_i operator.
6. Similarly, pick a representative $\overline{X}_i$ for each of the logical X operators for S. Let $\overline{X}_i' = \overline{X}_i M^{s(\overline{X}_i,P)}$. $\overline{X}_i'$ is a representative for the new logical X_i operator.

Procedure. We are given a circuit consisting of a m-element sequence of unitary Clifford group gates and Pauli measurements. At time step i, the circuit calls for either Clifford group gate $U_i \in \mathsf{C}_n$ (specified by its action on Paulis $U_i : X_j \mapsto U_i(X_j), Z_j \mapsto U_i(Z_j)$) or measurement of the eigenvalue of $P_i \in \mathsf{P}_n$ (assuming P_i has eigenvalues ± 1, not $\pm\iota$). The initial state of the circuit is given by stabilizer S, which encodes k qubits ($0 \leq k \leq n$), has generators $M_1, \ldots, M_{n-k}$, and has logical Pauli operators $\overline{X}_j$, $\overline{Z}_j$ ($j = 1, \ldots, k$).

1. **let** $k' \leftarrow k$
2. **let** $\mu_j = M_j$ ($j = 1, \ldots, n-k$)
3. **let** $\overline{\chi}_j \leftarrow \overline{X}_j$ ($j = 1, \ldots, k$)
4. **let** $\overline{\zeta}_j \leftarrow \overline{Z}_j$ ($j = 1, \ldots, k$)
5. **let** $|\psi\rangle \leftarrow$ encoded state of the system, corresponding to the logical state $\left|\overline{\psi}\right\rangle$
6. **for** $i = 1$ to m
    1. **let** $S \leftarrow \langle \mu_1, \ldots, \mu_{n-k} \rangle$ `// current stabilizer`
    2. **if** the circuit calls for Clifford gate U_i
        1. $\mu_j \leftarrow U_i(\mu_j)$ ($j = 1, \ldots, n-k'$)
        2. $\overline{\chi}_j \leftarrow U_i(\overline{\chi}_j)$ ($j = 1, \ldots, k'$)
        3. $\overline{\zeta}_j \leftarrow U_i(\overline{\zeta}_j)$ ($j = 1, \ldots, k'$)
        4. $S \leftarrow \langle \mu_1, \ldots, \mu_{n-k'} \rangle$
        5. If desired, choose a new set of generators for S (update the μ_js)
        6. If desired, rewrite $\overline{\chi}_j \leftarrow \mu \overline{\chi}_j$ or $\overline{\zeta}_j \leftarrow \mu \overline{\zeta}_j$ with $\mu \in S$
    3. **else if** the circuit calls for measurement of Pauli P_i
        1. **if** P_i does not commute with all μ_j `// this is the case` $P_i \notin \mathsf{N}(S)$
            1. Sample a uniformly random bit $b \in_R \{0, 1\}$
            2. **yield** measurement result $(-1)^b$
            3. Find $M \in \{\mu_j\}$ s.t. $\{M, P_i\} = 0$; reorder the generators s.t. $\mu_1 = M$
            4. $\mu_j \leftarrow \mu_j M^{s(\mu_j,P_i)}$ ($1 < j \leq n-k'$)
            5. $\mu_1 \leftarrow (-1)^b P$
            6. $S \leftarrow \langle \mu_1, \ldots, \mu_{n-k'} \rangle$; if desired, choose new generators for the revised S
            7. Pick a representative $\overline{\zeta}_j$ for each of the logical Z operators for S
            8. $\overline{\zeta}_j \leftarrow \overline{\zeta}_j M^{s(\overline{\zeta}_j,P_i)}$ `// representative for the new` $\overline{Z}_j$ `operator`
            9. Pick a representative $\overline{\chi}_j$ for each of the logical X operators for S
            10. $\overline{\chi}_j \leftarrow \overline{\chi}_j M^{s(\overline{\chi}_j,P_i)}$ `// representative for the new` $\overline{X}_j$ `operator`

2. **else if** $\pm P_i \notin S$ `// this is the case` $P_i \in \hat{\mathsf{N}}(S) \setminus \hat{S}$
   1. Write $P_i = \prod_{\ell=1}^{n-k'} \mu_j^{t_\ell} \prod_{r=1}^{k} \overline{\chi}_r^{u_r} \overline{\zeta}_r^{v_r}$, with $t_\ell, u_r, v_r \in \{0,1\}$
   2. **let** $Q \leftarrow \prod_{j=1}^{k} \overline{X}_j^{u_j} \overline{Z}_j^{v_j}$ `// the logical operator corresponding to` P
   3. Perform a measurement of Q on the current logical state $\left|\overline{\psi}\right\rangle$
   4. **yield** measurement result $(-1)^b$
   5. Update $|\psi\rangle$ accordingly `// may not be efficient depending on` $\left|\overline{\psi}\right\rangle$
   6. $k' \leftarrow k' - 1$ `// one logical qubit collapses (logical d.o.f.--)`
   7. $S \leftarrow \langle \mu_1, \ldots, \mu_{n-k'}, (-1)^b P \rangle$; if desired, choose new generators for S
   8. Update $\overline{\chi}_j$ and $\overline{\zeta}_j$ to relate to the new $|\psi\rangle$ `// may not be efficient`
3. **else if** $-P \in S$
   1. **yield** measurement result -1
4. **else** `// this is the case` $P \in S$
   1. **yield** measurement result $+1$

7. **return** k', $S = \langle \mu_1, \ldots, \mu_{n-k'} \rangle$, $(\overline{X}_j \mapsto \overline{\chi}_j, \overline{Z}_j \mapsto \overline{\zeta}_j)$

The output state lies in the stabilizer code $\mathsf{Q}(S)$. The encoded (logical) state has had $k - k'$ (logical) qubits measured, and has undergone the transformation $\overline{X}_j \mapsto \overline{\chi}_j$, $\overline{Z}_j \mapsto \overline{\zeta}_j$ $(j = 1, \ldots, n - k')$ on the remaining qubits.

The simulation of this procedure needs $O(n^3 m)$ time, taking into account all the linear algebraic manipulations needed to process measurement (in BSR). However, by keeping track of slightly more information, this can be reduced to $O(n^2 m)$.

Theorem. *Gottesman-Knill.* Given a circuit starting from an initial stabilizer state, followed by a sequence of Clifford group operations and Pauli measurements, which may depend on classical computations performed on previous measurement results, there is an efficient classical simulation of the circuit.

5.1.1 One-Bit Teleportation

A one-qubit teleportation circuit has two qubits as input: an arbitrary state $|\psi\rangle$ and an ancilla $|0\rangle$. The outputs of the circuit are a classical bit on the wire with the arbitrary state, and a qubit on the ancilla, such that the original state $|\psi\rangle$ can be recovered using the classical bit and the ancilla qubit. In effect, the arbitrary state $|\psi\rangle$ teleports from one wire to the other.

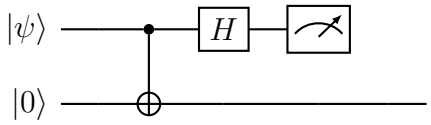

Figure 4: One-qubit teleportation: an example of a 2-qubit circuit made of Clifford group gates and measurement.

The input of this circuit is in the code generated by the stabilizer $S = \langle I \otimes Z \rangle$. The code encodes a single logical qubit. Representatives for the logical X and Z operations are

$\overline{X}_1 = X \otimes I$ and $\overline{Z}_1 = Z \otimes I$. The simulation of the circuit until after the Hadamard yields:

$$\begin{aligned}
\mu_1 = I \otimes Z &\xrightarrow{\text{CNOT}_{12}} Z \otimes Z \xrightarrow{H_1} X \otimes Z \\
\overline{\chi}_1 = X \otimes I &\xrightarrow{\text{CNOT}_{12}} X \otimes X \xrightarrow{H_1} Z \otimes X \\
\overline{\zeta}_1 = Z \otimes I &\xrightarrow{\text{CNOT}_{12}} Z \otimes I \xrightarrow{H_1} X \otimes I
\end{aligned}$$

At this point, we have a measurement on the first qubit. A measurement in a circuit without a specification means "measure in the computational basis," which is another way of saying by default, measure Z on the qubit. Hence, we measure the Pauli $P = Z \otimes I$. In this case, there is only one element of the stabilizer, and it anticommutes with P, which puts us in the case $P \notin \mathsf{N}(S)$.

1. Suppose we get the measurement outcome $+1$. We find the generator $M = \mu_1$ which anticommutes with P. We replace μ_1 with P. $\overline{\chi}_1$ commutes with P so its representative need not change. However, we can choose a new coset representative to take advantage of the new stabilizer: $(Z \otimes X)(Z \otimes I) = I \otimes X$. $\overline{\zeta}_1$ anticommutes with P, so we *must* choose a new coset representative

   $$\overline{\zeta}_1 \leftarrow \overline{\zeta}_1 M = (X \otimes I)(X \otimes Z) = I \otimes Z\,.$$

   After the measurement, the stabilizer and logical Paulis are thus

   $$\begin{aligned}
   \mu_1 &: Z \otimes I \\
   \overline{\chi}_1 &: I \otimes X \\
   \overline{\zeta}_1 &: I \otimes Z
   \end{aligned}$$

   We see that the input qubit has been moved to the second qubit for the output, and the output value of the first qubit is $|0\rangle$.

2. Suppose we get the measurement outcome -1. The generator M that commutes with P is still μ_1. μ_1 is now replaced with $-P$ instead of P. The new representative for $\overline{\chi}_1$ also acquires this minus sign since $\overline{\chi}_1 \leftarrow \overline{\chi}_1(-P)$. The new representative for $\overline{\zeta}_1$ does not acquire the minus sign since $\overline{\zeta}_1 \leftarrow \overline{\zeta}_1 M$. After the measurement, the stabilizer and logical Paulis are thus

   $$\begin{aligned}
   \mu_1 &: -Z \otimes I \\
   \overline{\chi}_1 &: -I \otimes X \\
   \overline{\zeta}_1 &: I \otimes Z
   \end{aligned}$$

   The output state of the first qubit is $|1\rangle$, and the output state of the second qubit is the input data qubit, but with a phase flip (Z) performed on it.

5.2 Generators of the Clifford Group

Theorem. Any gate in the n-qubit Clifford group C_n can be written as a product of $e^{\iota\theta} I$, H_i, $\sqrt{Z}_i$, and CNOT_{ij}, with $i, j = 1, \ldots, n$ and $\theta \in [0, 2\pi)$.

Proof.

1. The Pauli group is inside the group generated by H and $\sqrt{Z}$ since $Z = \sqrt{Z}^2$ and $X = HZH$. Global phases are covered by the gates $e^{\iota\theta} I$.

2. To finish the proof, we thus need only to show that the BSRs of H, $\sqrt{Z}$, and CNOT generate $\check{\mathsf{C}}_n$. It will be helpful to also use two additional gates: SWAP and CZ. Both are in the Clifford group since $\mathrm{SWAP}_{ij} = \mathrm{CNOT}_{ij}\mathrm{CNOT}_{ji}\mathrm{CNOT}_{ij}$ and $\mathrm{CZ} = (I \otimes H)\mathrm{CNOT}(I \otimes H)$.

3. In general if we have a symplectic matrix $U = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ representing an arbitrary Clifford group gate U, then we will find a sequence of H, $\sqrt{Z}$, and CNOT to multiply by on the left and the right to transform U into the identity:

$$\left(\prod_{\ell=1}^{n_L} V_{L,\ell}\right) U \left(\prod_{r=1}^{n_R} V_{R,r}\right) = I\,,$$

with the V_Ls and V_Rs drawn from $\{H_i, \sqrt{Z}_i, \mathrm{CNOT}_{ij}\}$. Then,

$$U = \left(\prod_{\ell=n_L}^{1} V_{L,\ell}^\dagger\right)\left(\prod_{r=n_R}^{1} V_{R,r}^\dagger\right),$$

proving the theorem.

Gate	Left multiplication	Right multiplication
H_i	Switches the ith rows of $(A\|B)$ and $(C\|D)$	Switches the ith columns of $\left(\frac{A}{C}\right)$ and $\left(\frac{B}{D}\right)$
$R_{\pi/4,i}$	Adds the ith row of $(A\|B)$ to the ith row of $(C\|D)$	Adds the ith column of $\left(\frac{B}{D}\right)$ to the ith column of $\left(\frac{A}{C}\right)$
$\mathrm{CNOT}_{i,j}$	Adds the ith row of $(A\|B)$ to the jth row of $(A\|B)$, and adds the jth row of $(C\|D)$ to the ith row of $(C\|D)$	Adds the jth column of $\left(\frac{A}{C}\right)$ to the ith column of $\left(\frac{A}{C}\right)$, and adds the ith column of $\left(\frac{B}{D}\right)$ to the jth column of $\left(\frac{B}{D}\right)$
$\mathrm{C-Z}_{i,j}$	Adds the ith row of $(A\|B)$ to the jth row of $(C\|D)$, and adds the jth row of $(A\|B)$ to the ith row of $(C\|D)$	Adds the jth column of $\left(\frac{B}{D}\right)$ to the ith column of $\left(\frac{A}{C}\right)$, and adds the ith column of $\left(\frac{B}{D}\right)$ to the jth column of $\left(\frac{A}{C}\right)$
$\mathrm{SWAP}_{i,j}$	Swaps the ith rows of A, B, C, and D with the jth rows of A, B, C, and D	Swaps the ith columns of A, B, C, and D with the jth columns of A, B, C, and D

Figure 5: The effects of left and right multiplication by H, $\sqrt{Z}$, CNOT, CZ, and SWAP on a symplectic matrix.

5. Let a_i, b_i, c_i, and d_i be the i^{th} columns of A, B, C, and D, respectively, and a_{ij}, b_{ij}, c_{ij}, and d_{ij} be the i, j^{th} entries of A, B, C, and D. Because U is symplectic, we get the following properties:

    1. U has full rank.
    2. $C^\mathsf{T}A + A^\mathsf{T}C = O$; i.e. $(a_i \| c_i) \odot (a_j \| c_j) = 0$.
    3. $C^\mathsf{T}B + A^\mathsf{T}D = I$; i.e. $(a_i \| c_i) \odot (b_j \| d_j) = \delta_{ij}$.
    4. $D^\mathsf{T}A + B^\mathsf{T}C = I$, which is the same as the previous property.
    5. $D^\mathsf{T}B + B^\mathsf{T}D = O$; i.e. $(b_i \| d_i) \odot (b_j \| d_j) = 0$.

    These properties all remain true if we multiply U on the left and right by other symplectic matrices. To find a sequence like the one we want, we can use the following steps:

    Procedure.

    1. Since U has maximum rank, somewhere in the first column of A or C is a 1. Using left multiplication by H and SWAP, we can put this 1 in the upper left corner.
    2. Do column reduction on the first column of A: Use left multiplication by CNOT to add the 1 in the upper left corner of A to any other row of A with a 1 in the first column.
    3. Do row reduction on the first row of A: Use right multiplication by CNOT to add the 1 in the upper left corner of A to any other column of A with a 1 in the first row.
    4. Using left multiplication by $\sqrt{Z}$ and/or CZ, we can similarly make the first column of C to be all 0s.
    5. At this point, we find that the first row of C has also bcome all 0s. This must be the case since
    $$0 = (a_1 \| c_1) \odot (a_j \| c_j) = a_1 \cdot c_j + c_1 \cdot a_j = c_{1j} + 0\,.$$
    6. Repeat steps 1 to 5 on the second row and column of A and C to make them all 0 except for a_{22}, which is 1. Continue with all other rows and column in sequence, until we have $A = I$ and $C = O$.
    7. At this point it follows that $D = I$:
    $$\delta_{ij} = (a_i \| c_i) \odot (b_j \| d_j) = a_i \cdot d_j + c_i \cdot b_j = d_{ij}\,.$$
    Also, B is symmetric:
    $$O = D^\mathsf{T}B + B^\mathsf{T}D = B + B^\mathsf{T}\,.$$
    8. Right multiply by H for every qubit, switching A and B, and switching C and D, so now $B = C = I$, $D = O$, and A is symmetric.

9. Right multiply by $\sqrt{Z}$ as needed to eliminate the diagonal of A. Right multiply by CZ to eliminate all other elements of A, leaving $A = O$. Since $D = O$, these operations do not change C.

10. Right multiply by H for every qubit, switching the left and right halves back again. We are left with $A = D = I$, $B = C = O$.

To eliminate all the 1s in a single row and column of A and C takes $O(n)$ gates. Doing this for all n rows and columns thus requires $O(n^2)$ gates. Steps 8 and 10 only require $O(n)$ gates, but step 8 could require one gate for each element of A, which is again $O(n^2)$. Thus, the overall number of gates used in the procedure is $O(n^2)$. It turns out that this can be slightly improved, allowing an arbitrary element of the Clifford group to be written as a product of $O\left(\frac{n^2}{\log n}\right)$ generators.

5.3 Encoding Circuits for Stabilizer Codes

Any stabilizer code can be encoded with a Clifford group gate. Therefore, techniques useful for decomposing a Clifford group unitary into a detailed quantum circuit are also useful for deriving encoding circuits for stabilizer codes. The key idea is that the procedure we used to map an arbitrary Clifford group operation U to the identity can be used to map an arbitrary Clifford group operation U to any other arbitrary Clifford group operation V by simply starting with the Clifford group operation $V^\dagger U$ and mapping that to the identity. This procedure is referred to as procedure ϱ in this subsection.

5.3.1 Encoding a Stabilizer Code with Unspecified Logical Paulis

The most straightforward case is when the stabilizer is given, but not the logical Paulis. If the original state, pre-encoding, is $|0\rangle^{\otimes(n-k)} \otimes |\psi\rangle$ (where $|\psi\rangle$ is a k-qubit state), its initial stabilizer is $\langle Z_1, \ldots, Z_{n-k}\rangle$. The final stabilizer S, post-encoding, also has $n-k$ generators $M_1, \ldots, M_{n-k}$. The choice of generators is not unique, of course, so we may pick any set of generators for S that we like. Then we must simply find some Clifford group circuit that maps $Z_i \mapsto M_i$ for $i = 1, \ldots, n-k$.

Procedure. Generate a list of gates using the steps given below.

1. Write the generators $M_1, \ldots, M_k$ of S in BSR. Produce the first $n-k$ columns of a symplectic matrix $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ as follows: the i^{th} column of A is $a_i = x_{M_i}$. The i^{th} column of C is $c_i = z_{M_i}$.

2. $(a_1 \| c_1)$ is a nonzero vector, so there is a 1 somewhere in the first column. Using left multiplication by H and SWAP, we can put this 1 in the upper left corner.

3. Do column reduction on the first column of A: use left multiplication by CNOT to add the 1 in the upper left corner of A to any other row of A with a 1 in the first column.

4. Using left multiplication by $\sqrt{Z}$ and/or CZ, we can similarly make the first column of C to be all 0s.

5. Do row reduction on the first row of A: replace generators of the stabilizer to add the 1 in the upper left corner of A to any other column of A with a 1 in the first row. (It does not matter if this step is performed before or after the previous step.)

6. At this point, we find that the first row of C has also become all 0s. This must be the case since
$$0 = (a_1 \| c_1) \odot (a_j \| c_j) = a_i \cdot c_j + c_1 \cdot a_j = c_{1j} + 0\,.$$

7. Repeat the previous steps on the second row and column of A and C to make them all 0 except for a_{22}, which is 1. Continue with all other rows and column in the sequence up to the column number $n-k$.

8. Perform H on qubits 1 through $n-k$ to switch A and C.

Take the inverse of this product of gates. The resulting Clifford group circuit will encode in some coset of S; i.e. the stabilizer will be almost correct, except that we may have some error syndrome different from the correct trivial syndrome. (Also note that the choice of stabilizer generators may be different from that which was originally given to us.) Calculate the action of the encoding circuit on Z_i for $i = 1, \ldots, n-k$. If for any i, the resulting generator M_i has the wrong sign, add X_i to the beginning of the encoding circuit. Note that the above procedure is a little simpler than ϱ since we don't care about what happens to Z_i for $i > n-k$, and to the X_is altogether. Also, the choice of generators of the stabilizer is not unique, so we may permute columns and add columns to each other within A and C without using any actual gates. This simplified procedure is referred to as $\varkappa$ for this subsection.

5.3.2 Encoding a Stabilizer with Specified Logical Paulis

The desired Clifford group operation now maps $Z_i \mapsto M_i$ for $i = 1, \ldots, n-k$, $Z_{n-k+j} \mapsto \overline{Z}_j$ for $j = 1, \ldots, k$, and $X_{n-k+j} \mapsto \overline{X}_j$ for $j = 1, \ldots, k$. There are two straightforward options:

1. Use procedure ϱ with fewer simplifications than procedure $\varkappa$.

2. Take advantage of the (simplified) procedure $\varkappa$ to find an encoding circuit for a stabilizer without specified logical Paulis and see what actual logical Paulis $\overline{P}'$ it produces. Determine the logical Clifford group operation that maps $\overline{P}'$ to the desired logical Pauli $\overline{P}$ using procedure ϱ on k qubits. Perform this circuit on qubits $n-k, \ldots, n$, followed by the encoding circuit given by procedure $\varkappa$. The resulting circuit will then encode the code with the correct logical Paulis.

For instance, an encoding circuit for the 5-qubit code might give the logical Paulis:

$$X_5 \to Z \otimes I \otimes I \otimes Z \otimes X$$
$$Z_5 \to Z \otimes Z \otimes Z \otimes Z \otimes Z$$

In this encoding, $\overline{Z}$ is correct. However,

$$\overline{X} = Z \otimes I \otimes I \otimes Z \otimes X = -(X \otimes I \otimes X \otimes Z \otimes Z)(Z \otimes X \otimes I \otimes X \otimes Z)(X \otimes X \otimes X \otimes X \otimes X)$$

Therefore, we need the Clifford operation $X \to -X, Z \to Z$. This is just the Clifford gate Z, so we can correct the circuit by beginning with a Z_5 gate.

5.4 Universal Gate Set

Since the Clifford group can be simulatd classically, to do any really interesting quantum algorithms, we will need some gates outside the Clifford group. how many gates and which ones suffice? It turns out *any* single gate will do it. Popular choices include CCNOT a.k.a. the Toffoli gate, or $\sqrt[4]{Z}$ a.k.a. the T gate.

Theorem. *Solovay-Kitaev.* Let $\mathsf{SU}(2^n)$ be the special unitary group, which is the set of all $2n \times 2n$ unitary matrices with determinant 1. Let $G \leq \mathsf{SU}(2^n)$ be a group of quantum gates such that $\hat{\mathsf{C}}_n \subsetneq G$. Then G is dense in $\mathsf{SU}(2^n)$; i.e. for any $U \in \mathsf{SU}(2^n)$ and any $\varepsilon > 0$, $\exists V \in G$ such that $\|U - V\|_o < \varepsilon$ (where $\|A\|_o$ is the operator/spectral norm defined as $\|A\|_o = \sup_{\|x\|_2=1} \|Ax\|_2$).

Thus, we can approximate all quantum gates to any desired degree of accuracy using G, even if G is generated by just the Clifford group, plus any additional gate.

6 Tighter, Please

Upper And Lower Bounds On Quantum Codes

6.1 Quantum Gilbert-Varshamov Bound

The main distinction from the classical GV bound is that there are more quantum errors that occur on a qubit than there are classical errors that occur on a bit.

Theorem. *Quantum Gilbert-Varshamov bound.* Suppose

$$\sum_{r=0}^{d-1} 3^r \binom{n}{r} \leq 2^{n-k} .$$

Then there exists an $((n, 2^k, d))$ QECC. Let $R = \frac{k}{n}$ be the rate, and $\delta = \frac{d}{n}$ be the relative distance, then for large n, a code exists if

$$R \leq 1 - \delta \lg 3 - H(\delta) ,$$

where

$$H(x) = -x \lg x - (1 - x) \lg(1 - x) .$$

Actually, we can show that an $[[n, k, d]]$ stabilizer code exists.

Proof.

1. Imagine making a long list of all $[[n, k]]$ stabilizer codes. We are going to run through all possible errors of weight up to $d - 1$. For each error E, we can check which codes can detect the error and which cannot, and cross off the list any code that can't detect E. Once we finish running through all the errors, we know that any code that remains must have distance at least d. We'll show that there must be at least one code to survive, giving us the desired $[[n, k, d]]$ code.

2. For any given error E, we need to count how many codes detect it. For any two errors $E, F \in \mathsf{P}_n \setminus \{I\}$, there exists some Clifford group element U with $U(E) = F$. Furthermore, if stabilizer code Q is a code that doesn't detect E, then $U(Q)$ is a stabilizer code that doesn't detect F. That is, U will permute the list of stabilizer codes and their sets of undetectable errors. Thus, the number of codes that cannot detect E must be the same as the number of codes that cannot detect F. That is, all non-identity errors in the Pauli group are undetectable for the same number ξ of $[[n, k]]$ codes.

3. Suppose there are β $[[n, k]]$ stabilizer codes. The code Q with stabilizer $S = \mathsf{S}(Q)$ detects the errors outside $\hat{\mathsf{N}}(S) \setminus \hat{S}$, which contains $2^{n+k} - 2^{n-k}$ elements. If we consider a bigger list of pairs (Q, E) for all $E \in \hat{\mathsf{N}}(\mathsf{S}(Q)) \setminus \hat{\mathsf{S}}(Q)$, we can count the number of elements on the list in two ways: as the number of errors times the number of codes that cannot detect each error, or as the number of codes times the number of errors that each code cannot detect. There are $4^n - 1$ non-identity errors, and I never appears in $\mathsf{N}(\mathsf{S}(Q)) \setminus \mathsf{S}(Q)$, so we have that:

$$(4^n - 1)\xi = \beta(2^{n+k} - 2^{n-k})\,.$$

4. For any non-identity error with weight less than d, there are ξ codes which do not detect it, so we can cross those off our original list. There are a total of $\tau - 1$ non-identity errors of weight less than d, with

$$\tau = \sum_{r=0}^{d-1} 3^r \binom{n}{r}\,.$$

   By the time we finish going through all errors of weight less than d, we have crossed off at most $(\tau - 1)\xi$ codes. If $(\tau - 1)\xi < \beta$, then at least one code remains.

5. Hence, the condition we get is

$$\frac{2^{n+k} - 2^{n-k}}{4^n - 1}(\tau - 1)\beta < \beta\,,$$

   or

$$\boxed{2^{n-k}(\tau - 1) < \frac{4^n - 1}{4^k - 1}}\,. \tag{1}$$

   This is the tightest version of the qGV bound, but notice that

$$\frac{4^n - 1}{4^k - 1} > \frac{4^n}{4^k} = 4^{n-k}\,,$$

   so if

$$2^{n-k}\tau \leq 4^{n-k} \tag{2}$$

   is satisfied, then equation (1) is satisfied too. Therefore, if equation (2) holds then an $[[n, k, d]]$ code exists. But this is exactly what we wanted to prove.

The proof is *non-constructive*. In this case, as with the classical GV bound, that means that, while the proof specifies a procedure to find a code with these parameters, the procedure is exponentially long as a function of n, so it's not useful in practice.

6.2 Quantum Hamming Bound

Theorem. *Quantum Hamming bound.* If a nondegenerate $((n, K, 2t+1))_q$ code exists, then

$$K\left(\sum_{r=0}^{t}(q^2-1)^r\binom{n}{r}\right) \leq q^n\,.$$

Proof. Because the code is nondegenerate, we know that the matrix $C_{ab} = \langle\psi|E_a^\dagger E_b|\psi\rangle$ (for any codeword $|\psi\rangle$) has maximum rank when $E_a^\dagger$ and E_b run over any basis for the space of $\leq t$-qudit errors. In particular, this means that the states $E_a|\psi\rangle$ are all linearly independent. Furthermore, if $|\psi\rangle$ and $|\phi\rangle$ are two orthogonal codewords, then the QECC conditions tell us that $\langle\psi|E_a^\dagger E_b|\psi\rangle = 0$, so E_a and E_b are orthogonal for any a, b. Thus if we let $\{|i\rangle\}$ be a basis for the code space, the set of states $\{E_a|i\rangle\}$ (over all a, i) are linearly independent. The code space has dimension K and there are q^2-1 one-qudit non-identity errors, so there are a total of

$$\tau = \sum_{r=0}^{t}(q^2-1)^r\binom{n}{r}$$

linearly independent errors of weight up to t. Thus we have a set of $K\tau$ linearly independent vectors, which we must fit into a Hilbert space of n qudits. Therefore, $K\tau < q^n$.

The big catch in the statement of the quantum Hamming bound is that while it provides an upper bound, the upper bound only holds for nondegenerate codes. Insisting that a code be degenerate is potentially a big limitation, yet it is easier to prove bounds on the existence of nondegenerate codes.

The quantum Hamming bound does not make essential use of the qudit structure of the space, or the fact that we are dealing with t-qudit errors. In general, if we have a nondegenerate QECC with a K-dimensional code space living in an N-dimensional physical Hilbert space, and the code can correct $\mathcal{E}$, a set of linearly independent error, then $K\,|\mathcal{E}| \leq N$.

6.3 Quantum Singleton Bound

Lemma. If an $((n, 1, d))_q$ QECC exists, then $n - 1 \geq 2(d-1)$.

Proof. If a code has distance d, then it can correct $d-1$ erasure errors. We can imagine dividing the n registers of the code into two sets, a set A with $d-1$ registers and a set B with $n-(d-1)$ registers. Set B has experienced $d-1$ erasures (because it is the remnant after "removing" set A), so using just those registers, it is possible to reconstruct the original encoded state $|\psi\rangle$. Now suppose that $n-1 < 2(d-1)$. Then $n-(d-1) < d-1$, and therefore set A has also experienced at most $d-1$ erasure errors. We would then be able to reconstruct a second copy of $|\psi\rangle$ using just the registers in set A. We could then use this code to clone an arbitrary state $|\psi\rangle$ as follows: encode $|\psi\rangle$ using the QECC, split up the registers into the sets A and B, and then reconstruct $|\psi\rangle$ from each set independently. This contradicts the no-cloning theorem.

Theorem. *Quantum Singleton bound.* If an $((n, q^k, d))_q$ QECC exists, then

$$n - k \geq 2(d-1)\,.$$

Proof sketch. For $k > 1$, split the registers into three sets A, B, and C, with A and B having $d-1$ registers each and C having $n-2(d-1)$ registers. Let R be a reference system with $\dim R = k$, and let RQ be in a maximally entangled state supported on the code space. Use additivity and subadditivity of the quantum Von Neumann entropy $S(\rho) = -\operatorname{tr}(\rho \lg \rho)$ to complete the argument.

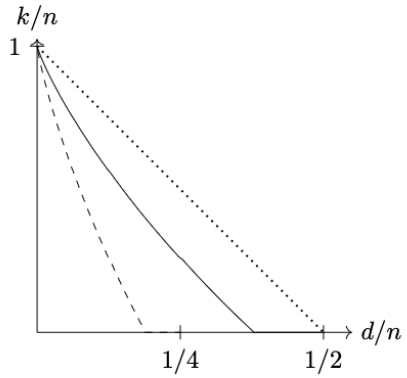

Figure 6: Quantum Hamming bound (solid), quantum GV bound (dashed), and quantum Singleton bound (dotted) for large n, $q = 2$.

6.4 Linear Programming Bounds

A linear programming bound provides a set of linear inequalities that a QECC with a given set of parameters $((n, K, d))_q$ must satisfy. If we can prove that the set of inequalities has no solutions for a given n, K, and d, then we know that a QECC with those parameters cannot exist. If there *is* a solution, then a QECC may or may not exist, but the linear programming solution gives us a number of constraints about the structure of any possible code with those parameters, so provides a good hint as to what to look at to find a code.

Definition. Let Π be the projector onto the code space of some $((n, K))$ QECC Q. Let

$$A_i = \frac{1}{K^2} \sum_{P \in \hat{\mathsf{P}}_n : \operatorname{wt}(P) = i} |\operatorname{tr}(P\Pi)|^2 \ .$$

The *weight enumerator* of Q is the degree n polynomial

$$A(x) = \sum_{i=0}^{n} A_i x^i \ .$$

Proposition. When Q is a stabilizer code with stabilizer $S = \mathsf{S}(Q)$,

$$A_i = |\{M \in S : \operatorname{wt} M = i\}| \ .$$

Definition. Let Π be the projector onto the code space of some $((n, K))$ QECC Q. Let

$$B_i = \frac{1}{K} \sum_{P \in \hat{\mathsf{P}}_n : \mathrm{wt}(P) = i} \mathrm{tr}(P \Pi P^\dagger \Pi) \,.$$

The *dual weight enumerator* of Q is the degree n polynomial

$$B(x) = \sum_{i=0}^{n} B_i x^i \,.$$

Proposition. When Q is a stabilizer code with stabilizer $S = \mathsf{S}(Q)$,

$$B_i = |\{N \in \mathsf{N}(S) : \mathrm{wt}\, N = i\}| \,.$$

Theorem. Let Q be a QECC with weight enumerator $A(x) = \sum A_i x^i$ and dual weight enumerator $B(x) = \sum B_i x^i$. Then

1) $A_0 = B_0 = 1$,

2) $B_i \geq A_i \geq 0$,

3) Q has distance d if and only if $A_i = B_i$ for $i < d$.

Definition. A $((n, K, d))$ QECC with weight enumerators $A(x)$ and $B(x)$ is *pure* if $A_i = B_i = 0$ for $i < d$. A code that is not pure is *impure*.

$\star$ For stabilizer codes, pure is the same as nondegenerate. For more general codes, degeneracy implies impurity.

Theorem. *Quantum MacWilliams identity.* Suppose we have an $((n, K))$ QECC with weight enumerator $A(x)$ and dual weight enumerator $B(x)$. Then

$$B(x) = \frac{K}{2^n} (1 + 3x)^n A\Big(\frac{1 - x}{1 + 3x}\Big) \,.$$

Definition. Suppose Π is the projector onto the code space of some $((n, K))$ QECC Q. Let

$$Sh_i = \frac{1}{K} \sum_{P \in \hat{\mathsf{P}}_n : \mathrm{wt}\, P = i} \mathrm{tr}(P \Pi P^\dagger Y^{\otimes n} \Pi^* Y^{\otimes n}) \,.$$

The *shadow enumerator* of Q is

$$Sh(x) = \sum_{i=0}^{n} Sh_i x^i \,.$$

Theorem. Suppose we have an $((n, K))$ QECC with weight enumerator $A(x)$ and shadow enumerator $Sh(x) = \sum Sh_i x^i$. Then

1) $Sh_i \geq 0 \ \forall j$,

2) $Sh(x)$ can be determined from $A(x)$:

$$Sh(x) = \frac{K}{2^n}(1+3x)^n A\Big(\frac{x-1}{1+3x}\Big)\,.$$

Notice that the relation between $A(x)$ and $Sh(x)$ differs from the MacWilliams identity only in that there is an $x-1$ in the numerator of the argument of A instead of $1-x$.

7 Bigger Can Be Better

Qudit Codes

7.1 Qudit Pauli Group

A *qudit* is a generic term for a q-dimensional Hilbert space, considered as a fundamental unit of the overall Hilbert space of dimension q^n.

7.1.1 Prime Dimension

Definition. The *single-qudit Pauli group* $\mathsf{P}_1(p)$ for prime $p > 2$ consists of elements $\{\omega^a X^b Z^c\}$, where

$$\begin{aligned}\omega &= e^{\frac{2\pi\iota}{p}} \text{ (the } p^{\text{th}} \text{ root of unity)}\\ X|j\rangle &= |(j+1) \bmod p\rangle \text{ (add one mod } p)\\ Z|j\rangle &= \omega^j|j\rangle \text{ (phase shift by } \omega)\,,\end{aligned}$$

and $0 \le a, b, c \le p-1$. The *n-qudit Pauli group* $\mathsf{P}_n(p) = \mathsf{P}_1(p)^{\otimes n}$: it consists of tensor products of n terms, each of the form $X^b Z^c$, with an overall phase factor of ω^a. $\hat{\mathsf{P}}_n(p) = \mathsf{P}_n(p)/\{\omega^a I\}$ is the qudit Pauli group without phases.

$\star$ $(X^a Z^b)(X^c Z^d) = \omega^{bc-ad}(X^c Z^d)(X^a Z^b)\,.$

$\star$ We are no longer choosing between commuting and anti-commuting Paulis; now commutation is measured by an integer modulo p, which describes the phase factor that appears when we move P past Q as a power of ω. Since ω is a p^{th} root of unity, we might as well calculate $bc-ad$ using arithmetic modulo p. We can define a new function $s(P,Q) : \mathsf{P}_n(p) \times \mathsf{P}_n(p) \to \mathbb{Z}_p$ by

$$PQ = \omega^{s(P,Q)} QP\,.$$

$\star$ All elements of $\mathsf{P}_n(p)$ have order p. This means that the eigenvalues of all elements of the qudit Pauli group are of the form ω^j for integer j.

$\star$ We can define a $\mathbb{Z}_p$ symplectuc representation of the qudit Pauli group by

$$P = \bigotimes_{j=1}^{n} X^{a_j} Z^{b_j} \mapsto v_P = (x_P \| z_P)\,,$$

with x_P and z_P being n-component vectors over $\mathbb{Z}_p$. x_P has entries a_j and z_P has entries b_j. The overall phase of P is lost when moving to the symplectic representation. Suppose v_Q is the symplectic representation of $Q = \otimes_{j=1}^{n} X^{c_j} Z^{d_j}$, we can define a symplectic inner product between v_P and v_Q as

$$v_P \odot v_Q = \sum_{j=1}^{n} b_j c_j - a_j d_j \,,$$

with arithmetic taken mod p.

Proposition. Let $P, Q \in \mathsf{P}_n(p)$. Then $s(P,Q) = v_P \odot v_Q$.

We can also map the qudit Pauli group to $\mathrm{GF}(p^2)$. We should think of $\mathrm{GF}(p^2)$ as a 2-dimensional vector space over $\mathrm{GF}(p)$ with basis $\{1, \alpha\}$. α can be any element of $\mathrm{GF}(p^2)$ that is not in the $\mathrm{GF}(p)$ subfield. We can write arbitrary $\beta \in \mathrm{GF}(p^2)$ as

$$\beta = a + b\alpha \,,$$

with $a, b \in \mathrm{GF}(p)$. We then map

$$X^a Z^b \mapsto a + b\alpha \,.$$

We lose the overall phase, so if we start with an element of $\hat{\mathsf{P}}_n(p)$, we end up with an n-component vector over $\mathrm{GF}(p^2)$.

Definition. Suppose a and b are n-dimensional vectors over $\mathrm{GF}(p^2)$. Let

$$a * b = \frac{a \cdot b^{\odot p} - b \cdot a^{\odot p}}{\alpha - \alpha^p} \,,$$

where $\odot$, here, represents the Hadamard product between vectors, and $\cdot$ represents the usual inner product between vectors. This is also called a *trace-alternating inner product*.

Theorem. Suppose $P, Q \in \hat{\mathsf{P}}_n$ correspond to vectors a, b over $\mathrm{GF}(p^2)$. Then, $a * b = s(P,Q)$.

7.1.2 Prime Power Dimension

Suppose each qudit has dimension $q = p^m$, with p a prime. The most staightforward thing to do is to let $\mathsf{P}_n(q) = \mathsf{P}_m(p)^{\otimes n}$. In other words, we consider each q-dimensional qudit as broken up into m p-dimensional qudits.

Definition. Suppose $\beta \in \mathrm{GF}(q)$. Let X^β and Z^β be defined as

$$X^\beta |\gamma\rangle = |\gamma + \beta\rangle$$
$$Z^\beta |\gamma\rangle = \omega^{\mathrm{tr}(\beta\gamma)} |\gamma\rangle \,,$$

where $\gamma \in \mathrm{GF}(q)$, $\omega = e^{\frac{2\pi\iota}{p}}$ is a p^{th} root of unity, and Tr is the $\mathrm{GF}(q)$ trace function which maps elements of $\mathrm{GF}(q)$ to elements of $\mathrm{GF}(p)$. Then $\mathsf{P}_n(q)$ consists of elements of the form

$$\eta \omega^a \bigotimes_{j=1}^{n} X^{\beta_j} Z^{\gamma_j} \,,$$

where $a \in \mathrm{GF}(p)$, and $\beta_j, \gamma_j \in \mathrm{GF}(q)$. For odd q, η is always 1. For even q, η can be 1 or ι. As usual,

$$\hat{\mathsf{P}}_n(q) = \begin{cases} \mathsf{P}_n(q)/\{\iota^a I\}, & p = 2 \\ \mathsf{P}_n(q)/\{\omega^a I\}, & p \text{ odd} \end{cases} .$$

$\star$ $Z^\gamma X^\beta = \omega^{\mathrm{tr}(\gamma\beta)} X^\beta Z^\gamma$.

Suppose we consider $\mathrm{GF}(q)$ as an m-dimensional vector space over $\mathrm{GF}(p)$, so

$$\gamma = \sum_{i=0}^{m-1} a_i \alpha_i \,,$$

with $\gamma \in \mathrm{GF}(q)$, $a_i \in \mathrm{GF}(p)$, and the α_is elements of $\mathrm{GF}(q)$ which are linearly independent vectors when considered over $\mathrm{GF}(p)$. It is most common to take $\alpha_i = \alpha^i$, with α some primitive element of $\mathrm{GF}(q)$. Note that with this choice, $\alpha_0 = 1$. Hence, $|\gamma\rangle \leftrightarrow |a_0\rangle \otimes |a_1\rangle \otimes \ldots \otimes |a_{m-1}\rangle$.

- X^{α_i} then has a natural interpretation as the p-dimensional X applied to the i^{th} tensor factor:

  $$X^{\alpha_i}|\gamma\rangle = |\gamma + \alpha_i\rangle = \left|\sum_{j=0}^{m-1} a_j\alpha_j + \alpha_i\right\rangle = \left|\sum_{j=1}^{m-1} (a_j + \delta_{ij})\alpha_j\right\rangle .$$

  If we apply $X^{\sum b_i\alpha_i}$ instead, that corresponds to performing X^{b_i} in the i^{th} tensor factor. That is, to understand the action of X^β, we expand β in the same basis used for the standard basis decomposition, and apply the appropriate power of X on each factor.

- The set $\{\alpha_i\}$ forms a basis for $\mathrm{GF}(q)$ considered as an m-dimensional vector space over $\mathrm{GF}(p)$, and for any basis, there exists a dual basis $\{\beta_j\}$ with the property:

  $$\mathrm{tr}(\alpha_i\beta_j) = \delta_{ij} \,.$$

  Then,

  $$Z^{\beta_j}|\gamma\rangle = \omega^{\mathrm{tr}\left(\beta_j \sum_i a_i\alpha_i\right)} \bigotimes_{i=0}^{m-1} |a_i\rangle = \omega^{a_j} \bigotimes_{i=0}^{m-1} |a_i\rangle \,,$$

  since trace is linear over $\mathrm{GF}(p)$. That is, Z^{β_j} corresponds to performing the p-dimensional Z on the j^{th} tensor factor. To understand the action of Z^β in general, we simply then expand β in the dual basis to the one used for th standard basis.

The choice of the dual bases $\{\alpha_i\}$ and $\{\beta_j\}$ thus specifies an isomorphism $\mathsf{P}_n(q) \cong \mathsf{P}_{mn}(p)$.

When we drop phases from the Pauli group, we get vectors on a symplectic space:

$$P = \bigotimes_{j=1}^{n} X^{\eta_j} Z^{\tau_j} \mapsto v_P = (x_P \| z_P) \,,$$

with x_P an n-dimensional vector over $\mathrm{GF}(q)$ with entries η_j, and z_P an n-dimensional vector with entries τ_j. The symplectic inner product between v_P and v_Q (with $Q = \bigotimes_{j=1}^n X^{\mu_j} Z^{\xi_j}$) is

$$v_P \odot v_Q = \sum_{j=1}^{n} \mathrm{tr}(\tau_j\mu_j - \eta_j\xi_j) \,.$$

To map the q-dimensional Pauli group into $\mathrm{GF}(q^2)$, we pick an element $\alpha \in \mathrm{GF}(q^2) \setminus \mathrm{GF}(q)$, and map $(x\|z) \mapsto x + \alpha z$. The correct symplectic inner product in this case is

$$a * b = \mathrm{tr}_{\frac{q}{p}}\left(\frac{a \cdot b^{\odot q} - b \cdot a^{\odot q}}{\alpha - \alpha^q}\right),$$

where $\odot$, here, represents the Hadamard product between vectors, and $\cdot$ represents the inner product between vectors. Note that both the symplectic inner products $\odot$ and $*$ give elements in $\mathrm{GF}(p)$ bcause of the trace operation. We want elements in $\mathrm{GF}(p)$ because the phase factor ω^a is used to encode whether two Paulis commute, where ω is a p^{th} root of unity.

Theorem. Suppose $P, Q \in \hat{\mathsf{P}}_n(q)$ correspond to vectors a, b over $\mathrm{GF}(q^2)$. Then,

$$a * b = s(P, Q)\,.$$

7.1.3 Other Dimensions

If the dimension q of the register is not a prime power, there are two sensible ways to generalize the above approaches to define a Pauli group.

1. One idea is to break q up into its prime factorization $q = \prod p_i^{\alpha_i}$, and treat each prime power factor as a separate sub-register of size $p_i^{\alpha_i}$ with its own Pauli group.

2. The second idea is to directly generalize the Pauli group used for prime dimensions, by letting

$$\begin{aligned} \omega &= e^{\frac{2\pi\iota}{q}} \\ X|j\rangle &= |(j+1) \bmod q\rangle \\ Z|j\rangle &= \omega^j |j\rangle\,. \end{aligned}$$

   As usual, the n-qudit Pauli group consists of products of the form $\omega^a \bigotimes_{i=1}^n X^{b_i} Z^{c_i}$ for odd q. For even q, we must include a possible overall factor of ι as well. This version of the Pauli group is known as the *Heisenberg-Weyl group*. It is also possible to use the Heisenberg-Weyl group for prime power dimensions.

7.1.4 Nice Error Bases

Definition. In a q-dimensional Hilbert space, let $\mathcal{E} = \{E_1, \ldots, E_{q^2}\}$ be a set of unitary operators satisfying $E_1 = I$ and $\mathrm{tr}\, E_i^\dagger E_j = q\delta_{ij}$. The set $\mathcal{E}$ is a *nice error basis* if $E_i E_j = \omega_{ij} E_{f(i,j)}$ for all i, j and some phases ω_{ij}.

To get a generalized Pauli group from a nice error basis, we take elements of the form ωE_i, where ω is a product of ω_{ij}s.

Proposition. The indices i of the errors in a nice error basis form a group under multiplication given by the binary operation $f(i, j)$.

Definition. The group of indices $\{i\}$ under group operation $f(i, j)$ is called the *index group* of the nice error basis.

The index group can also be obtained by taking the Pauli group generated by the nice error basis and modding out overall phase. For the usual qubit Pauli group, the index group is therefore isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$. For the Heisenberg-Weyl group in dimension q, the index group is $\mathbb{Z}_q \times \mathbb{Z}_q$, and for the prime power Pauli group, the index group is $\mathrm{GF}(q) \times \mathrm{GF}(q)$ under addition. All of these are fairly straightforward Abelian groups, but for large q, there exist error groups with more exotic index groups, including non-Abelian ones.

7.2 Qudit Stabilizer Codes

Definition. Let $q = p^m$ be a prime power, and let Q be a subspace of the Hilbert space $\mathcal{H}_q^{\otimes n}$ (i.e., consisting of n qudits). The $\mathrm{GF}(q)$ *stabilizer of* Q is the set

$$\mathsf{S}(Q) = \{M \in \mathsf{P}_n(q) : |\psi\rangle \text{ is an eigenvector of } M \text{ with eigenvalue } +1 \ \forall |\psi\rangle \in Q\}.$$

Let S be a subgroup of $\mathsf{P}_n(q)$. We say that S is a $\mathrm{GF}(q)$ *stabilizer*, or just stabilizer, when the $\mathrm{GF}(q)$ is clear from context, if it is Abelian, and if $e^{\iota\phi} I \notin S$ for any $\phi \neq 0$. The *code space* of a $\mathrm{GF}(q)$ stabilizer S is the subspace

$$\mathsf{Q}(S) = \{|\psi\rangle : M|\psi\rangle = |\psi\rangle \, \forall M \in S\}.$$

The code Q is a $\mathrm{GF}(q)$ *stabilizer code* iff $Q = \mathsf{Q}(\mathsf{S}(Q))$. The *normalizer* $\mathsf{N}(S)$ of the stabilizer S is

$$\mathsf{N}(S) = \{N \in \mathsf{P}_n(q) : NM = MN \, \forall M \in S\}.$$

The only real differences from the qubit definition are the use of $\mathsf{P}_n(q)$ instead of P_n and forbidding $e^{\iota\phi}$ for all nonzero phases ϕ, which is needed because there are more phases in $\mathsf{P}_n(q)$ than just $\pm 1, \pm\iota$.

Definition. Let $P \in \mathsf{P}_n(q)$ have $\mathrm{GF}(q)$ symplectic representation $(x_P \| z_P)$. Let S be a $\mathrm{GF}(q)$ stabilizer, with symplectic representation $\overline{S}$. Then we say $\overline{S}$ is a *true* $\mathrm{GF}(q)$ *stabilizer* if $\forall \gamma \in \mathrm{GF}(q)$, $(x_P \| z_P) \in \overline{S} \implies (\gamma x_P \| \gamma z_P) \in \overline{S}$. That is, for a true $\mathrm{GF}(q)$ stabilizer code, the symplectic representation of the stabilizer is a $\mathrm{GF}(q)$-linear space.

Note that if q is prime, any $\mathrm{GF}(q)$ stabilizer code is a true $\mathrm{GF}(q)$ stabilizer code since $P \in S$ implies $P^i \in S$ as well, and $(x_{P^i} \| z_{P^i}) = i(x_P \| z_P)$.

Proposition. If Q is a non-trivial subspace of the HIlbert space, the $\mathsf{S}(Q)$ is a $\mathrm{GF}(q)$ stabilizer (not necessarily true). If S is a $\mathrm{GF}(q)$ stabilizer, then $\mathsf{S}(\mathsf{Q}(S)) = S$.

Theorem. Let p be a prime and let S be a $\mathrm{GF}(p)$ stabilizer for the code $\mathsf{Q}(S)$, which has n physical qudits. If $|S| = p^r$ (i.e., S has r generators), then $\dim \mathsf{Q}(S) = p^{n-r}$, so $\mathsf{Q}(S)$ encodes $k = n - r$ logical qudits. The set of undetectable errors for $\mathsf{Q}(S)$ is $\hat{\mathsf{N}}(S) \setminus \hat{S}$. The distance of $\mathsf{Q}(S)$ is $d = \min_{E \in \hat{\mathsf{N}}(S) \setminus \hat{S}} \mathrm{wt}\, E$.

Notation. A stabilizer code with n physical qudits of dimension q, k logical qudits (also of dimension q), and distance d is denoted as an $[[n, k, d]]_q$ code.

7.3 Qudit Calderbank-Shor-Steane Codes

Theorem. Let C_1 and C_2 be two classical linear codes over GF(q) with parameters $[n, k_1, d_1]_q$ and $[n, k_2, d_2]_q$ and satisfying $C_1^\perp \subseteq C_2$. Then there exists a true GF(q) stabilizer code with the stabilizer given as follows:

1) For prime dimension, generate a stabilizer code of the form

$$\begin{bmatrix} O & H_1 \\ H_2 & O \end{bmatrix},$$

from the parity check matrices of C_1 and C_2. The only difference from the qubit case is that the arithmetic to determine commutation is mod p instead of mod 2.

2) For prime power qudits, define the stabilizer S to be the smallest group containing all $\bigotimes_j Z^{\nu_j}$ and $\bigotimes_j X^{\eta_j}$, where ν runs over elements of $C_1^\perp$ and η runs over elements of $C_2^\perp$. (Note that is is *not* sufficient to look at the group generated in this way for the basis vectors ν and η of $C_1^\perp$ and $C_2^\perp$. This is because if $\nu \in C_1^\perp$, then GF(q) linearity implies that $\xi\nu \in C_1^\perp$, but $\bigotimes_j Z^{\nu_j} \in S$ is not sufficient to imply that $\bigotimes_j Z^{\xi\nu_j} \in S$.

This stabilizer has parameters $[[n, k_1 + k_2 - n, d]]_q$, with $d \geq \min(d_1, d_2)$.

$\star$ As with binary CSS codes, the basis codewords have a straightforward form when written out in the standard basis:

$$\left|\nu + C_2^\perp\right\rangle = \sum_{\eta \in C_2^\perp} |\nu + \eta\rangle,$$

for $\nu \in C_1$. This works for both prime dimension and prime power dimension. Indeed, we could just take it as the definition of a CSS code in any dimension.

7.4 Polynomial Codes

One particularly interesting family of qudit CSS codes is the family of *polynomial codes*, which are CSS codes derived from classical Reed-Solomon codes or their variants. Let us pick the code C_1 as a Reed-Solomon code by choosing n distinct points $(\alpha_1, \ldots, \alpha_n)$ from GF(q) $\setminus \{0\}$ ($n < q$) and a number $k_1 < n$. We'll use polynomials with degree up to $k_1 - 1$. We also pick a second number $0 \leq k_2 \leq k_1$ which will determine the size of C_2.

Definition. The basis codewords of the polynomial code over GF(q) with (n, k_1, k_2) are

$$\left|\overline{\beta_0, \ldots, \beta_{k_2-1}}\right\rangle = \sum_{\beta_{k_2}, \ldots, \beta_{k_1-1} \in \mathrm{GF}(q)} \bigotimes_{i=1}^{n} \left|\beta_0 + \beta_1\alpha_i + \beta_2\alpha_i^2 + \ldots + \beta_{k_1-1}\alpha_i^{k_1-1}\right\rangle.$$

The polynomial code is the span of these basis codewords.

A qudit polynomial code is a qudit CSS code since the basis codewords have the correct form. However, instead of choosing both C_1 and C_2 to be Reed-Solomon codes (which wouldn't work since they are not precisely dual to each other), we have chosen $C_2^\perp$ to be a modified Reed-Solomon code. In particular, $C_2^\perp$ is the code made up of vectors $(f(\alpha_1), \ldots, f(\alpha_n))$, where $f(x)$ runs over degree $k_1 - 1$ or lower polynomials for which the lowest nonzero term

is the x^{k_2} power or higher. With this choice, it is manifestly true that the code encodes k_2 qudits.

Theorem. The GF(q) polynomial code with parameters (n, k_1, k_2) is a nondegenerate true $[[n, k_2, d]]_q$ stabilizer code with $d = \min(n - k_1 + 1, k_1 - k_2 + 1)$.

7.5 Qudit Clifford Group

Definition. Let p be a prime. The *qudit Clifford group* is

$$\mathsf{C}_n(p) = \{U \in \mathsf{U}(p^n) : UPU^\dagger \in \mathsf{P}_n(p)\, \forall P \in \mathsf{P}_n(p)\}\,.$$

As with qubits, $\mathsf{P}_n(p)$ is a normal subgroup of $\mathsf{C}_n(p)$. We can also define

$$\hat{\mathsf{C}}_n(p) = \mathsf{C}_n(p)/\{e^{\iota\theta} I\}$$
$$\check{\mathsf{C}}_n(p) = \hat{\mathsf{C}}_n(p)/\hat{\mathsf{P}}_n(p)\,.$$

$\star$ $\mathsf{C}_n(p) \cong \mathsf{Sp}(2n, \mathbb{Z}_p)$.

$\star$ There exists a unitary in $\mathsf{C}_n(p)$ that performs any transformation of the Paulis that preserves commutation relations.

$\star$ It is possible to efficiently simulate the behavior of a qudit Clifford group, including Pauli measurements, with a classical computer using a procedure analogous to the simulation of the qubit Clifford group.

The main difference in the simulation is when measuring P_i with $P_i \notin \mathsf{N}(S)$. The measurement outcome is a uniform random value $b \in_R \{0, \ldots, p-1\}$ rather than 0 or 1. We still find a generator M of the stabilizer to replace with $\omega^b P_i$, but we choose M that has a factor of ω when commuting past P_i rather than anticommuting with it. By taking the appropriate power r of M, we can ensure that NM^r commutes with P for stabilizer generators or logical operators N; we replace the original N with the appropriate NM^r.

The biggest difference with the qubit Clifford group comes when comparing the actual elements:

1. The discrete Fourier transform over $\mathbb{Z}_p$ is the generalization of the Hadamard transform (which is the discrete Fourier transform over $\mathbb{Z}_2$):

$$\mathcal{F}|a\rangle = \frac{1}{\sqrt{p}} \sum_{b=0}^{p-1} \omega^{ab} |b\rangle\,.$$

   Working out the conjugation action on the Paulis, we find that $\mathcal{F}$ does not precisely swap X and Z, but has the following action:

$$X \mapsto Z$$
$$Z \mapsto X^{-1}\,.$$

2. The two-qudit SUM gate is the direct analogue of the qubit CNOT gate:

$$\begin{aligned}\mathrm{SUM}|a\rangle|b\rangle &= |a\rangle|a+b \bmod p\rangle\\ X\otimes I &\mapsto X\otimes X\\ I\otimes X &\mapsto I\otimes X\\ Z\otimes I &\mapsto Z\otimes I\\ I\otimes Z &\mapsto Z^{-1}\otimes Z\,.\end{aligned}$$

   Similar to the action of the Fourier transform, there is an extra inverse in one of the images in the conjugation action of SUM. The inverse powers are needed to ensure that the Clifford group gate preserves commutation relations among the Paulis. For instance, $Z\otimes Z$ does not commute with $X\otimes X$, but $Z^{-1}\otimes Z$ does.

3. There are operators in $\mathsf{C}_n(p)$ which don't have any qubit analogue, such as scalar multiplication by $\alpha\in\mathbb{Z}_p\setminus\{0\}$:

$$\begin{aligned}S_\alpha|a\rangle &\mapsto |\alpha a \bmod p\rangle\\ X &\mapsto X^\alpha\\ Z &\mapsto Z^{\alpha^{-1}}\,.\end{aligned}$$

4. There is no precise qudit analogue of $\sqrt{Z}$. The closest is a quadratic phase gate B which has a similar action on Paulis:

$$\begin{aligned}B|a\rangle &= \omega^{\frac{a(a-1)}{2}}|a\rangle\\ X &\mapsto XZ\\ Z &\mapsto Z\,.\end{aligned}$$

5. $\mathrm{SWAP}_{ij} = S_{-1_i}\mathrm{SUM}_{ij}\mathrm{SUM}_{ji}^{-1}\mathrm{SUM}_{ij}$.

6. The analogue of CZ is $(I\otimes\mathcal{F})\mathrm{SUM}(I\otimes\mathcal{F})$.

Theorem. The gates $\mathcal{F}$, B, and SUM, along with global phase $e^{\iota\theta}I$ generate $\mathsf{C}_n(p)$.

8 Now, What Did I Leave Out?

Other Things You Should Know About Quantum Error Correction

8.1 Concatenated Codes

If one QECC is good, two must be better.

Definition. Let Q be an $((n_1,K,d_1))_{q_1}$ code qith encoder $\mathfrak{E}_1$ and R be an $((n_2,q_1,d_2))_{q_2}$ code with encoder $\mathfrak{E}_2$. The *concatenated code* formed from Q and R is an $((n_1n_2,K))_{q_2}$ code with encoder $\mathfrak{E}_2^{\otimes n_1}\circ\mathfrak{E}_1$. Q is known as the *inner code* and R is known as the *outer code*.

Proposition. The concatenated code formed from a distance d_1 inner code and a distance d_2 outer code has distance $d \geq d_1 d_2$.

Definition. Let $Q_1 = Q$ be an $((n, q, d))_q$ code, and let Q_k be the $((n^k, q, d^k))_q$ code obtained by concatenating Q_{k-1} as the inner code with Q as the outer code. We say that Q_k is a code involving k *levels* of concatenation. The physical qudits form *level 0 qudits*, the outer code for Q_k is *level 1* of concatenation, and its logical qudits are *level 1 qudits*. The logical qudits for the outer code for Q_ℓ that appears in the recursive definition of Q_k are *level* $k - \ell + 1$ *qudits* and the logical qudits for the overall code are *level* k *qudits*.

If we concatenate two stabilizer codes, the resulting code is a stabilizer code as well, with the stabilizer constructed in a particular way:

Procedure. Let the inner code be an $[[n_1, 1, d_1]]$ code with stabilizer S_1, and let the outer code be an $[[n_2, k, d_2]]$ code with stabilizer S_2 and logical Paulis $\overline{P}$. Then the stabilizer S of the concatenated code formed from these two codes is given as follows:

1) For each generator $M \in S_2$, include in S the Pauli M_i consisting of M acting on the i^{th} block of n_2 qubits in the code tensored with the identity on all other blocks.

2) For each generator $M \in S_1$, replace every single-qubit Pauli P_i in its tensor product decomposition (i.e., acting on the i^{th} physical qubit) with $\overline{P}_i$, the corresponding logical Pauli from the outer code acting on the i^{th} block of n_2 qubits in the concatenated code. Take the tensor product of all the $\overline{P}_i$ operators for a single $M \in S_1$ and include that in S.

3) Form the group generated by the elements in the previous steps.

One can also determine the logical Paulis for the concatenated code by the same process as in step 2: take a logical Pauli for S_1 and replace each Pauli in its tensor product decomposition with the corresponding logical Pauli for S_2.

⋆ Concatenated stabilizer codes will often be degenerate.

8.1.1 Decoding Concatenated Codes

Procedure. To decode a concatenation of Q and R, first correct and decode each block of R separately. The resulting qudits form a codeword of Q, possibly with logical errors inherited from the blocks of R. Then correct and decode Q.

This is efficient whenever the constituent decoders are efficient, but need not exploit the full distance of the concatenated code. For example, concatenating the five-qubit code three times gives an $[[125, 1, 27]]$ code, which can correct 13 arbitrary errors. If we first decode its 25 five-qubit blocks separately and then decode the remaining 25-qubit code, two errors in each of five blocks can already put five logical errors into the 25-qubit code, beyond its guaranteed correction capability. This uses only 10 physical errors.

⋆ Decoding the levels separately throws away useful information. A nonzero syndrome in a small block tells us that its decoded qubit is less trustworthy. Passing this information to the

next decoder can improve performance. It is not quite an erasure flag, however, because a logical error can also occur with zero syndrome.

If an $[[n_1, k_1, d_1]]_{2^{k_2}}$ code is concatenated with an $[[n_2, k_2, d_2]]_2$ code, the result is a qubit code with parameters $[[n_1 n_2, k_1 k_2, d \geq d_1 d_2]]_2$. Thus, qudit codes can be useful in constructing qubit codes.

For ℓ levels of an $[[n_0, 1, d_0]]$ code, the fractional distance is

$$\frac{d}{n} = \left(\frac{d_0}{n_0}\right)^{\ell},$$

which decreases with ℓ when $d_0 < n_0$. Nevertheless, the code can perform very well against typical independent errors. Let $t = \left\lfloor \frac{d_0 - 1}{2} \right\rfloor \geq 1$, and suppose each physical qubit independently suffers a Pauli error with probability p. For small p, estimating a block failure by $t+1$ erroneous qubits gives

$$p_1 \approx \binom{n_0}{t+1} p^{t+1} = p_T \left(\frac{p}{p_T}\right)^{t+1}, \qquad p_T = \binom{n_0}{t+1}^{-1/t}.$$

Iterating the level-by-level decoder gives

$$p_\ell \approx p_T \left(\frac{p}{p_T}\right)^{(t+1)^\ell}.$$

For $p < p_T$, the logical error rate rapidly approaches zero. The estimate assumes all patterns of $t+1$ errors cause failure; correcting some of them improves the result.

$\star$ A decreasing fractional distance describes worst-case errors, not typical independent errors. The patterns that defeat a concatenated decoder with relatively few errors must be clustered in particular ways, and can be very unlikely. This calculation also assumes perfect encoding and decoding; noisy operations require fault tolerance.

8.2 Convolutional Codes

A block code encodes a fixed number of logical qubits into a fixed number of physical qubits. A convolutional code instead allows the stream to be extended, adding logical qubits and encoding them without having to revisit all the physical qubits already produced. The encoder keeps a finite quantum memory and releases the earlier qubits to the channel.

Definition. A *qubit quantum convolutional code* is a family of $((n_i, 2^{k_i}))$ codes Q_i, $i \in \mathbb{Z}^+$, with the following properties:

1) There are fixed positive integers r, s, t with $s \leq r \leq t$.

2) $n_i = n_{i-1} + r$ and $k_i = k_{i-1} + s$ for $i > 1$.

3) There are t-qubit unitaries U_i, V_i for $i \geq 2$ and an n_1-qubit unitary W such that the encoder for Q_i is

$$V_i U_i U_{i-1} \cdots U_2 W .$$

Here W acts on the first k_1 logical qubits and $n_1 - k_1$ ancillas in $|0\rangle$. Each U_i acts on physical positions $n_i - t + 1, \ldots, n_i$, taking the last $t - r$ qubits retained from the previous step, s new logical qubits, and $r - s$ new $|0\rangle$ ancillas. V_i acts on the final t qubits to terminate the stream.

The *rate* of the convolutional code is $s/r = \lim_{i\to\infty} k_i/n_i$.

W starts the encoder and V_i finishes it. The repeated work is done by the U_i. At each step, r new physical positions are added and only a fixed-size memory remains involved in future encoding. Most often the U_i and V_i are identical up to shifts. For a stabilizer convolutional code, the encoding unitaries are Clifford operations.

$\star$ Finite memory is the meaningful quantum condition. We cannot simply require that encoding a new qubit leaves all previous quantum information untouched: even a controlled gate can change both systems when its inputs are superposed or entangled.

8.2.1 Shift-Invariant Stabilizers

For a shift-invariant stabilizer convolutional code, the bulk stabilizer is generated by repeated shifts of $r - s$ generator patterns. There can be additional generators at the beginning and end. We will focus on generators supported on a fixed finite interval, independently of the total stream length.

Notation. Let D denote a shift by r physical qubits. If $M_{1,j}$ is a generator pattern, write

$$M_{i,j} = D^{i-1} M_{1,j} \, .$$

For coefficients $\alpha_i \in \mathbb{Z}_2$, define

$$\left(\sum_i \alpha_i D^i \right) (M) = \prod_i (D^i M)^{\alpha_i} \, .$$

Thus, addition of polynomial terms represents multiplication of shifted Paulis, with overall phases handled separately.

In binary symplectic notation, write

$$v_P(D) = (x_P(D) \mid z_P(D)) \, ,$$

where $x_P(D)$ and $z_P(D)$ each have r polynomial entries. The coefficient of D^i describes the X or Z component in frame i. Finite-support operators use polynomials, or Laurent polynomials if negative shifts are included. Formal infinite series are useful for an unterminated stream, but a finite code still requires boundary conditions.

Proposition. All shifts $D^i P$ and $D^j Q$ commute if and only if

$$x_P(D) \cdot z_Q(D^{-1}) + z_P(D) \cdot x_Q(D^{-1}) = 0 \, ,$$

as a Laurent polynomial over $\mathbb{Z}_2$.

Proof. Write $x_P(D) = \sum_a x_{P,a} D^a$ and similarly for the other components. The coefficient of D^k in the displayed expression is

$$\sum_a (x_{P,a} \cdot z_{Q,a-k} + z_{P,a} \cdot x_{Q,a-k}) \,,$$

which is the binary symplectic product of P with $D^k Q$. It vanishes precisely when those operators commute. Since shifting both operators leaves their commutation relation unchanged, checking every coefficient checks every relative shift.

$\star$ It is not enough that the generators in one frame commute. They must also commute with every overlapping shifted copy, including shifted copies of themselves. The D^{-1} in the condition keeps track of these relative shifts.

Example. One rate-$\frac{1}{5}$ convolutional code, based on the five-qubit code, has the bulk generator patterns

$$\begin{aligned} M_1(D) &= X \otimes Z \otimes Z \otimes X \otimes I \\ M_2(D) &= I \otimes X \otimes Z \otimes Z \otimes X \\ M_3(D) &= XD \otimes I \otimes X \otimes Z \otimes Z \\ M_4(D) &= ZD \otimes XD \otimes I \otimes X \otimes Z \,. \end{aligned}$$

Here XD in the first slot means an X in the first position of the next five-qubit frame. The full bulk stabilizer includes all shifts of these four patterns. Initial and terminal generators complete a finite code.

8.2.2 Quantum Viterbi Algorithm

The encoder has $O(n)$ circuit size because it performs a constant-size operation for each fixed-size group of new qubits. For stabilizer generators of bounded spatial extent, syndrome decoding can also be linear in n.

Suppose every generator introduced at a step is supported within the last w physical positions, with $w \geq r$. To extend a candidate error across the next r qubits, we only need to remember its Pauli pattern on the preceding $w-r$ qubits. Earlier errors cannot affect a newly introduced check.

Algorithm. *Quantum Viterbi algorithm.* Given a syndrome, find a most likely Pauli error for an independent Pauli noise model, or a minimum-weight Pauli error consistent with that syndrome.

1) Make a table indexed by the 4^{w-r} Pauli patterns on the trailing $w-r$ positions, ignoring phase. For each pattern, store the best compatible partial error and its probability or weight. Initialize with all possible errors on the first $w-r$ qubits, rejecting any that disagree with checks already wholly supported there.

2) Extend each surviving partial error by every Pauli pattern on the next r qubits. Reject extensions that disagree with any syndrome bit whose generator is now fully contained in the processed region.

3) Multiply the old probability by the probability of the newly added pattern, or add its weight. Group the surviving extensions by their Pauli pattern on the new trailing $w - r$ positions and retain only the best candidate for each pattern.

4) Advance by r positions and repeat. At the final boundary, impose the remaining termination checks and choose the best surviving candidate. Trace back the stored choices to recover the full error.

Proof sketch. Two partial errors with the same trailing pattern affect all future checks in the same way. Any allowed continuation of one is therefore allowed for the other. Under independent noise, the continuation multiplies both probabilities by the same factor, or adds the same weight. The worse partial candidate can never become the better full candidate. Thus, retaining one best candidate per trailing pattern loses no optimum.

For fixed r, w, the number of table entries and transitions per step is constant, so decoding takes $O(n)$ time. Store predecessor pointers rather than copying the entire partial error at each update to retain this scaling.

$\star$ Bounded *weight* alone is not enough for this argument. The checks must have bounded *extent*: a weight-two check connecting qubits arbitrarily far apart would require information from arbitrarily far back in the stream.

$\star$ The most likely individual Pauli need not belong to the most likely logical error class for a degenerate code. Optimal logical decoding sums probabilities over stabilizer-equivalent errors. The algorithm above optimizes an individual error, or its weight, as stated.

8.2.3 Distance and Catastrophic Decoding

Proposition. A quantum convolutional code with fixed finite-memory parameters has distance bounded by a constant independent of the stream length. This holds even without shift invariance, Clifford encoding, or bounded-support stabilizer generators.

Proof sketch. Consider a sufficiently long interval of newly introduced logical qubits, with the logical qubits outside it fixed. Only the finite memory can transmit information from this interval to physical qubits far beyond it. Similarly, the earlier physical qubits can be correlated with the interval only through a fixed-size memory at its beginning. If the physical interval could be erased and corrected, all its logical information would have to be recoverable through these bounded-size boundary systems. Choosing more logical qubits than the boundaries can hold contradicts the erasure correction conditions. The size of such an interval depends on the memory parameters, not on the total stream length.

Thus, a fixed convolutional code cannot correct arbitrary errors at an increasing distance merely by extending the stream. Even independent noise will eventually produce a sufficiently dense local cluster. We would nevertheless like such a cluster to affect only a bounded number of logical qubits.

Definition. Suppose a decoder for a stabilizer convolutional code assigns a Pauli P_σ to syndrome σ. For an actual Pauli error Q, the residual operator

$$R_Q = P_{\sigma(Q)}^\dagger Q$$

has trivial syndrome and hence acts as a logical Pauli. The decoder is *catastrophic* if some finite-weight physical Pauli error gives a residual logical Pauli of unbounded logical weight as the stream length increases. A code is catastrophic if all its decoders are catastrophic.

⋆ The weight in the conclusion is the number of *logical* qubits changed. A non-catastrophic decoder may fail near an uncorrectable cluster, but it prevents that local failure from damaging an unbounded portion of the decoded stream.

8.3 Information-Theoretic Approach to QECCs

The Knill-Laflamme conditions describe correctability for a set of possible error operators. An information-theoretic condition instead starts with a specified noise channel and asks whether the channel preserves the quantum information in the encoded state. This viewpoint also generalizes naturally to approximate error correction, where the recovered state need only be close to the original.

8.3.1 Coherent Information

Let R be an isolated reference system and Q the system we retain. Introduce an environment E so that the joint state on RQE is pure. If RQ is pure, the entanglement entropy $S(Q) = S(R)$ measures the quantum information shared with the reference. When Q is also entangled with E, some of its entropy instead reflects correlations with the environment.

Definition. The *coherent information* in Q relative to R, for a joint state ρ_{RQ}, is

$$I_c(\rho_{RQ}) = S(\rho_Q) - S(\rho_{RQ}) = -S(R \mid Q)\,.$$

Here S is the von Neumann entropy, with logarithms base 2. If $\mathcal{E} : L \to Q$ is a channel and ρ_{RL} is a pure input state, its coherent information for that input is

$$I_c(\mathcal{E}, \rho) = S(\mathcal{E}(\rho_L)) - S((\mathcal{I}_R \otimes \mathcal{E})(\rho))\,.$$

For a pure joint output on RQE, this is also

$$I_c(\rho_{RQ}) = S(Q) - S(E)\,.$$

If $Q = Q_1 \otimes Q_2$, with Q_1 entangled only with R and Q_2 entangled only with E, then $I_c = S(Q_1)$, as expected.

⋆ Coherent information can be negative. Classical conditional entropy is nonnegative, so positive coherent information is an inherently quantum phenomenon. For a maximally entangled pair of n-qubit systems, $I_c = n$; if Q is completely uncorrelated with R, then $I_c = -S(R)$.

Definition. Let $V : L \to Q \otimes E$ be a Stinespring isometry for a channel $\mathcal{E}$, so that

$$\mathcal{E}(\rho) = \operatorname*{tr}_E(V \rho V^\dagger)\,.$$

Its *complementary channel* is

$$\widehat{\mathcal{E}}(\rho) = \operatorname*{tr}_Q(V \rho V^\dagger)\,.$$

It describes the information sent to the environment rather than to the receiver.

8.3.2 Quantum Data Processing Inequality

Theorem. *Quantum data processing inequality.* Let ρ_{RL} be any joint state and let $\mathcal{E} : L \to Q$ be a channel acting only on L. Then

$$I_c((\mathcal{I}_R \otimes \mathcal{E})(\rho)) \leq I_c(\rho)\,.$$

Equivalently, for successive channels $\mathcal{E}$ and $\mathcal{F}$,

$$I_c(\mathcal{F} \circ \mathcal{E}, \rho) \leq I_c(\mathcal{E}, \rho)\,.$$

Proof. Purify the initial state using E, and purify the channel using a fresh environment E'. The initial state on RLE and final state on $RQEE'$ are pure. Thus,

$$I_c(\rho) = S(L) - S(RL) = S(RE) - S(E)$$
$$I_c(\mathcal{E}, \rho) = S(Q) - S(RQ) = S(REE') - S(EE')\,.$$

Strong subadditivity gives

$$S(REE') + S(E) \leq S(RE) + S(EE')\,,$$

which is precisely the desired inequality after rearranging.

$\star$ A recovery channel cannot increase coherent information that has already been lost to the environment. Successful error correction preserves the logical information through the noise; it does not recreate destroyed information.

8.3.3 Coherent Information and Correctability

Theorem. Let $\mathcal{F} : L \to Q$ be a channel and let R have dimension at least that of L. There exists a CPTP decoding map $\mathcal{G}$ with

$$\mathcal{G} \circ \mathcal{F} = \mathcal{I}_L$$

if and only if

$$I_c(\mathcal{F}, \rho) = S(\rho_L)$$

for every pure state ρ on $R \otimes L$.

For a QECC, take $\mathcal{F} = \mathcal{E} \circ \mathfrak{E}$, the encoding isometry followed by the noise channel. The theorem is then an information-theoretic version of the Knill-Laflamme conditions for the Kraus operators of $\mathcal{E}$.

Proof.

1) If $\mathcal{G} \circ \mathcal{F} = \mathcal{I}$, data processing gives

$$S(\rho_L) = I_c(\mathcal{G} \circ \mathcal{F}, \rho) \leq I_c(\mathcal{F}, \rho) \leq I_c(\rho) = S(\rho_L)\,.$$

Equality must hold throughout.

2) Conversely, purify $\mathcal{F}$ by an isometry $V : L \to Q \otimes E$ and write the pure output state as σ_{RQE}. Since R is untouched and the input is pure,

$$S(\rho_L) = S(R), \qquad I_c(\mathcal{F}, \rho) = S(RE) - S(E)\,.$$

Preservation of coherent information therefore gives

$$S(RE) = S(R) + S(E)\,,$$

which holds if and only if $\sigma_{RE} = \sigma_R \otimes \sigma_E$.

3) In particular, choose a maximally entangled input on $R \otimes L$. By projecting the reference onto different states, we can prepare any pure input on L. These projections commute with V, which does not act on R. Since the final RE state is a product, the environment state is unchanged by the choice of input.

4) Regard V as an encoder and the discarded environment E as an erased subsystem. The erasure correction condition says that E is correctable precisely when its reduced state contains no logical information. This is what step 3 establishes, so a decoder $\mathcal{G}$ exists.

Corollary. It is enough to verify the coherent information condition for one maximally entangled input

$$|\Phi\rangle_{RL} = \frac{1}{\sqrt{K}} \sum_{i=1}^{K} |i\rangle_R |i\rangle_L, \qquad K = \dim L\,.$$

If $I_c(\mathcal{F}, |\Phi\rangle\langle\Phi|) = \log_2 K$, the channel is exactly reversible on L.

Corollary. A channel $\mathcal{F}$ is exactly reversible on the logical input space if and only if its complementary channel is constant there:

$$\widehat{\mathcal{F}}(\rho_L) = \sigma_E$$

for every logical density matrix ρ_L, with the same state σ_E.

⋆ For a pure input and a purified channel, the coherent information lost is exactly the mutual information acquired by the environment about the reference:

$$S(\rho_L) - I_c(\mathcal{F}, \rho) = S(R) + S(E) - S(RE) = I(R : E)\,.$$

Thus, preserving coherent information, decoupling the reference from the environment, and correcting the channel are three descriptions of the same condition.

II. Fault-Tolerant Quantum Computation

9 Everyone Makes Mistakes

Basics Of Fault Tolerance

Definition. Consider a quantum circuit, arranged into time steps. Each qubit can be involved in at most one action per time step, but mutltiple actions can be performed in a single time, provided they involve different qubits. A *location* is a single indivisible action in the circuit. We may consider the following different types of locations:

1) A *preparation location:* The preparation of a single qubit in a particular state, frequently $|0\rangle$. Sometimes we distinguish between different types of preparation locations based on the state prepared. Occasionally we may want a preparation location to encompass the creation of a multiple-qubit entangled state.

2) A *gate location:* A single gate from the universal gate set used to describe the circuit. Sometimes we may wat to distinguish different types of gate locations based on the kind of gate appearing (e.g., a CNOT location or a H location). If the gate is a multiple-qubit gate, the location encompasses all the qubits involved in the gate, but still just counts as a single location. Usually, we only consider unitary gates to produce gate locations, but one could also consider gate locations for more general quantum channels.

3) A *wait location* or *storage location:* A single qubit is stored for a unit time with no gate performed on it.

4) A *measurement location:* A single qubit is measured with a von Neumann measurement, usually in the standard computational basis. The measurement outcome is captured as classical information. Occasionally we may want locations corresponding to other kinds of measurements, including possibly multiple-qubit entangled measurements.

5) A *classical computation location:* In the basic model for fault tolerance, we omit locations corresponding to classical computations. We assume that classical computation can be performed instantaneously (and flawlessly) on the outcomes of measurements, including combining the results of multiple measurement locations. The outcomes of the classical computations may later be used to control any other locations later in the circuit.

Definition. Consider a quantum circuit C divided into locations, as above. An *independent error model* on the quantum circuit C is a circuit $\widetilde{C}$ where every preparation location, gate location, storage location, and measurement location is replaced by a separate quantum channel, arranged in the same way as the original circuit. In an *independent stochastic error model*, the quantum channel corresponding to a particular location L of C performs the correct action for L with probability $1 - p_L$ and does something else with the same input and output Hilbert space dimensions as L with probability p_L; i.e., the channel for a $|0\rangle$ preparation location L will prepare $|0\rangle$ with probability $1 - p_L$ and prepare some other single-qubit state (possibly mixed) with probability p_L. The quantum channel corresponding to L is called a *location* and may be referred to as $\widetilde{L}$, or just L if the distinction between the two is not needed. p_L is called the *error probability* or *error rate* of location L. The circuit $\widetilde{C}$ is called a *noisy* implementation of C. In any particular realization of a noisy circuit $\widetilde{C}$ with an independent stochastic error model, for some locations the correct action L is performed, while for other locations the wrong action is performed. The locations where the wrong thing happens are called *faulty*, or they are said to have a *fault*. We frequently say of a fault location that a preparation error, gate error, storage error, or measurement error has occurred, depending on th type of location. In an *independent Pauli error model*, a

faulty location always performs the correct action followed by a Pauli channel. In a Pauli error model, a fault measurement location will flip the measurement outcome bit with some probability. A *fault path* is a set of locations in C. A *Pauli fault path* is a fault path with a Pauli error associated with each location in it.

Definition. A *basic model for fault tolerance* is as follows: An ideal circuit C is physically realized as a noisy circuit $\widetilde{C}$, according to an independent stochastic error model. The error model has the property that all locations of a given type have the same error probability. In other respects, the quantum channels used in the error model are arbitrary, and may be incompletely specified. A basic error model can therefore be summarized in terms of four error probabilities: p_P (the error probability for a preparation location), p_G (the error probability for a gate location), p_S (the error probability for a storage location), and p_M (the error probability for a measurement location).

What do we want out of a fault-tolerant protocol? The basic idea is this: We are given a circuit, broken down into a universal set of gates. We'd like to replace the qubits of the circuit with the logical qubits of a QECC. Then we'd like to perform a sequence of gates on the qubits, without ever leaving the protection of the code, that transforms the logical qubits with the same unitary transformation as the original ideal circuit we were given. And we'd like the sequence of gates we use to be fault tolerant, in the sense that the errors on a small but constant fraction of the gates do not interfere with us getting the correct outcome for the logical qubits.

Definition. Given a particular type L of location, specified precisely (e.g., not just that it is a gate location, but also the type of gate), a *gadget for* L is a QECC Q coupled with a circuit G_L, such that if the qubits involved in L are encoded into Q, then subjected to the circuit G_L, then decoded, all without errors, the effect is the same as if L were performed directly. The circuit G_L may involve adding and/or discarding physical qubits, which may or may not also be encoded in Q. In other words, a gadget for a specific function is supposed to perform that function on the encoded state. If the encoder for Q is $\mathfrak{E}$ and the decoder is $\mathfrak{D}$, then $\mathfrak{D} \circ G_L \circ \mathfrak{E} = L$.

Definition. A *fault-tolerant protocol* consists of a QECC Q along with the following types of gadgets:

1) State preparation gadget for at least one type of state.

2) Measurement gadget for at least one type of measurement.

3) Gate gadgets for a universal set of gates.

4) Storage gadget.

5) Error correction gadget.

Definition. Let C be a circuit, broken down into locations. Then $\mathsf{FT}(C)$, the *ideal fault-tolerant simulation of* C associated with a given fault-tolerant protocol, is the circuit created as follows:

1) Take each location in C and replace it with the corresponding gadget of the fault-tolerant protocol, in the process replacing each qubit of C with a block of the QECC Q.

2) After each state preparation gadget, gate gadget, or storage gadget, place an error correction gadget on each of the blocks of Q involved in the gadget.

The *fault-tolerant simulation of* C is then $\widetilde{\mathsf{FT}(C)}$. The *circuit size overhead* of $\mathsf{FT}(C)$ is the number of locations in $\mathsf{FT}(C)$ divided by the number of locations in C. The *space overhead* or *qubit overhead* is the ratio of the number of physical qubits involved in $\mathsf{FT}(C)$ to the number of qubits involved in C; and the *depth overhead* or *time overhead* is the ratio of the depth of $\mathsf{FT}(C)$ to the depth of C.

We don't need error correction gadgets after measurement gadgets, because the output of a measurement gadget is just classical information, and we are assuming that classical circuits don't suffer errors.

9.1 Ideal Decoder and r-Filter

Definition. The *r-filter* for the QECC Q is the projector on the subspace spanned by states of the form $E|\psi\rangle$, where $|\psi\rangle \in Q$ and E is an error of weight at most r. Notationally, I will denote an r-filter by $\mathscr{F}_r$ (the book uses pictures).

When we apply an r-filter to some state, we get a state which contains only errors of weight less than or equal to r. The r-filter is thus a test we can apply to see if there are many errors on the state (relative to any codeword). If an input state passes through an r-filter unchanged, i.e., if $\mathscr{F}_r = \mathscr{I}$, then it has r or fewer errors on it.

Definition. The *ideal decoder* for the QECC Q is the quantum channel that takes a state (possibly with errors) encoded in Q, corrects the errors, and then decodes the logical state, discarding all ancilla qubits. I will denote an ideal decoder by $\mathscr{D}$.

9.2 Fault-Tolerant Gate Properties

Instead of the pictorial language the book uses, I use a notational language, using which we'll define properties that a fault-tolerant gate gadget must satisfy. For the purposes of these definitions, we can include storage gadgets as gate gadgets by thinking of them as performing the identity gate.

Notation. The notation to represent a noisy gate gadget with exactly s faults is $\mathscr{G}_s$ or $\mathscr{G}_s^2$ for one- and two-qubit gates, respectively. The input to these gates is a single block of the QECC. The notation for an ideal gate performed on unencoded qubits without noise is $\mathcal{G}$ or $\mathcal{G}^2$ for one- or two-qubit gates, respectively.

In an independent stochastic error model, these "pictures" can be interpreted as CP maps. For some particular realization of the noisy gate gadget circuit, there are some number s of faults specified by a fault path S. Replacing the locations in S with the CP maps corresponding to faults in the noise model (usually we'll want to rescale to make the maps TP, if possible) gives a linear map corresponding to the "picture." The "picture" for a gadget

with s faults thus represents many different maps, depending on the exact locations which are faulty and on the precise error model. The ideal decoder and r-filter are also CP maps; the decoder is TP, but the r-filter is not. We can compose them with CP maps, and the properties for the fault tolerance will be defined as equations relating CP maps derived this way. We can thus represent each equation simply as "pictures."

Definition. *GPP.* Suppose we have a gate gadget associated with a QECC with distance $2t+1$ (i.e. it corrects t errors). if it is a single-qubit gate gadget, it satisfies the *FT gate error propagation property (GPP)* if, whenever

$$\mathscr{G}_s \circ \mathscr{F}_r = \mathscr{F}_{r+s} \circ \mathscr{G}_s \circ \mathscr{F}_r \,.$$

A two-qubit gate gadget satisfies GPP if, whenever $r_1 + r_2 + s \leq t$,

$$\mathscr{G}_s^2 \circ \mathscr{F}_{r_1}\mathscr{F}_{r_2} = \mathscr{F}_{r_1+r_2+s}\mathscr{F}_{r_1+r_2+s} \circ \mathscr{G}_s^2 \circ \mathscr{F}_{r_1}\mathscr{F}_{r_2} \,.$$

For both equations, the faulty locations on both sides of the equation have the same fault path and same error maps substituted in.

These equations use filters on the input to ensure that the input state to the CP maps can be considered to be at most r errors away from a codeword. The equations then demand that, given such an input state, the state exiting the gadget be at most $r+s$ or r_1+r_2+s errors away from a codeword. In the case of the two-qubit gate gadget, we allow errors to propagate between the two blocks of code. Thus, even if $s=0$, so the gadget itself is not faulty, the number of errors in a block may increase as a result of error propagation. However, we insist that the number of errors ending up in a single block is limited by the total number of errors r_1+r_2 input in the two blocks plus the number s of new faults. This is the sense in which this property limits the error propagation. Note that for the two-qubit gate gadget, the condition uses separate filters on the two blocks of the code involved in the gadget. Thus, the final state of the two blocks taken together might have a total of $2(r_1+r_2+s)$ errors on it.

For the GPP, we only impose the equations when the total number of errors on input states plus faults in the gadget is at most t. When there are more than t total errors floating around, we don't in general insist that out gadgets have to work correctly. In many cases, fault-tolerant gate constructions do continue to satisfy these equations even when there are more errors, but that is an additional property not required by the basic definition.

Definition. *GCP.* Suppose we have agate gadget associated with a QECC with distance $2t+1$. If it is a single-qubit gate gadget, it satisfies the *FT gate correctness property (GCP)* if, whenever $r+s \leq t$,

$$\mathscr{D} \circ \mathscr{G}_s \circ \mathscr{F}_r = \mathcal{G} \circ \mathscr{D} \circ \mathscr{F}_r \,.$$

A two-qubit gate gadget satisfies the GCP if, whenever $r_1 + r_2 + s \leq t$,

$$\mathscr{D}\mathscr{D} \circ \mathscr{G}_s^2 \circ \mathscr{F}_{r_1}\mathscr{F}_{r_2} = \mathcal{G}^2 \circ \mathscr{D}\mathscr{D} \circ \mathscr{F}_{r_1}\mathscr{F}_{r_2} \,.$$

The equations must hold for any choice of CP maps substituted in for the faulty locations on the LHS. These conditions say that if we perform the noisy gate gadget and then decode, we get the same thing as if we decode and then perform the gate gadget.

The GCP ensures that a noisy gate gadget performs the correct operation on the encoded state, generalizing the definition of a gadget to the case in which there are errors. Again, we only insist that the conditions hold when the total number of errors floating around is at most t. In this case, that is all we can hope for.

9.3 Fault-Tolerant Preparation and Measurement Properties

Notation. I denote a single-qubit state preparation gadget with exactly s faults by $\mathscr{P}_s$. I denote a single-qubit measurement gadget with exactly s faults by $\mathscr{M}_s$. The book uses pictures. I denote the ideal single-qubit state preparation by $\mathcal{P}$, and the ideal single-qubit state measurement by $\mathcal{M}$.

Both the ideal measurement and the measurement gadget output classical information, generally a single bit. State preparation and measurement are CP maps. A noisy preparation gadget is a CP map that takes no input and outputs a state in the Hilbert space of the QECC. A noisy measurement gadget is a CP map that takes a noisy codeword as input and outputs a classical bit.

Definition. *PPP.* Suppose we have a preparation gadget associated with a QECC with distance $2t+1$. The gadget satisfies the *FT preparation error propagation property (PPP)* if, whenever $s \leq t$,

$$\mathscr{P}_s = \mathscr{F}_s \circ \mathscr{P}_s \,.$$

Definition. *PCP.* Suppose we have a preparation gadget associated with a QECC with distance $2t+1$. The gadget satisfies the *FT preparation correctness property (PCP)* if, whenever $s \leq t$,

$$\mathscr{D} \circ \mathscr{P}_s = \mathcal{P} \,.$$

In other words, when $s \leq t$, a preparation gadget with s faults in it should output a state with no more than s errors in it, and the state should be an encoding of the correct state. For the measurement gadget, we only have one property. We don't need to worry about error propagation through a measurement gadget because the output only involves classical information, which we are assuming is error-free.

Definition. *MCP.* Suppose we have a measurement gadget associated with a QECC with distance $2t+1$. The gadget satisfies the *FT measurement correctness property (MCP)* if, whenever $r+s \leq t$,

$$\mathscr{M}_s \circ \mathscr{F}_r = \mathcal{M} \circ \mathscr{D} \circ \mathscr{F}_r \,.$$

That is, if the inital state has no more than r errors, and there are not too many errors in the measurement gadget, the classical outcome of the measurement should be the same as if we did an ideal measurement on the decoded state.

9.4 Fault-Tolerant Error Correction Properties

Fault-tolerant error correction gadgets (FTECs) are a bit different from gadgets associated with locations. For other gadgets, we are concerned that error propagation is not too severe,

but FTECs are supposed to actually *reduce* the number of errors in the state. Of course, if there are faults in the FTEC itself, it may introduce more errors that must be corrected in a later FTEC. FTECs invariably involve adding extra ancilla qubits and produce syndrome information. The syndrome information may either be in the form of classical bits or as extra qubits, and is usually discarded at the end of the gadget.

Notation. I denote an FTEC with exactly s faults by $\mathscr{E}_s$.

Definition. *ERP.* Suppose we have an FTEC associated with a QECC with distance $2t+1$. The gadget satisfies the *FT error correction recovery property (ERP)* if, whenever $s \leq t$,

$$\mathscr{E}_s = \mathscr{F}_s \circ \mathscr{E}_s \,.$$

This says that no matter how many errors there are to start with, we want our FTEC to return us to a codeword (or a codeword with s faults when there are s faults in the FTEC gadget).

Definition. *ECP.* Suppose we have an FTEC associated with a QECC with distance $2t+1$. The gadget satisfies the *FT error correction correctness property (ECP)* if, whenever $r+s \leq t$,

$$\mathscr{D} \circ \mathscr{E}_s \circ \mathscr{F}_r = \mathscr{D} \circ \mathscr{F}_r \,.$$

This says that when the total number of errors (input errors plus faults in the gadget) is at most t, the encoded state does not change during an FTEC gadget.

9.5 Fault Tolerance Properties for Different Error Models

Theorem. Consider a realization C_1 of a noisy gadget circuit with a fault path of size s, and suppose that the location of each fault path is replaced by a linear Hilbert space operator (e.g. specific Kraus operators of the CP maps). Replace one particular fault location L in C_1 with a different linear Hilbert space operator, and call the new circuit C_2. Suppose the operator for L in C_1 is E and the operator for L in C_2 is F, and let C_3 be the circuit realization C_1 with L replaced by $\alpha E + \beta F$, for any $\alpha, \beta \in \mathbb{C}$. If the gadget satisfies the relevant fault tolerance properties for realizations C_1 and C_2, then it also satisfies them for C_3.

As a consequence of this, we need not check the fault tolerance properties for arbitrary CP maps inserted for faults. It is sufficient to check them for a basis of errors. In particular, Pauli errors suffice.

Corollary. If a gadget satisfies the relevant fault tolerant properties when arbitrary Pauli errors are inserted for all faults, then it satisfies the fault tolerance properties when arbitrary CP maps are inserted for faults.

Theorem. Consider a realization of a gadget with a particular fault path S, $|S| = s$. Suppose a gadget satisfies the relevant fault tolerance properties when arbitrary Pauli errors are inserted at the locations of S. Then it also satisfies the fault tolerance properties when subjected to a persistent environment which starts in an arbitrary state (possibly even one entangled with the qubits involved in the gadget), and interacts with the locations of S through arbitrary unitaries.

9.6 Threshold Theorem for the Basic Model

The threshold theorem is one of the key results in the theory of quantum information. If it were not true, it would probably be impossible to build big quantum computers.

Theorem. *Threshold theorem for the basic model of fault tolerance.* Suppose a system satisfies a basic model for fault tolerance, with $p_P = p_G = p_S = p_M = p$. Then there exists a family of fault-tolerant protocols $\mathfrak{F}_\ell$ and a threshold error rate p_T such that, if $p < p_T$, then for any ideal quantum circuit C which starts with preparation locations and ends with measurement locations, and any $\varepsilon > 0$, there exists an ℓ such that the output distribution of $\widetilde{\mathsf{FT}_\ell(C)}$, the noisy fault-tolerant simulation of C by $\mathfrak{F}_\ell$, has statistical distance at most ε from the output of C. When C has T locations, then $\mathsf{FT}_\ell(C)$ has $O\Big(T \operatorname{polylog}\Big(\frac{T}{\varepsilon}\Big)\Big)$ locations (i.e., the circuit size overhead is $O\Big(\operatorname{polylog}\Big(\frac{T}{\varepsilon}\Big)\Big)$).

In other words, if the physical error rates for all locations fall below the magical threshold error value p_T, there exists a fault-tolerant protocol that makes the logical error rate on the complete circuit as small as we like. Note that we actually need a *family* of FT protocols to make this work—as the circuit gets bigger, or the logical error rate we'd like to achieve gets smaller, then we need to work harder at getting rid of errors, which means more overhead and a more involved FT protocol. The nice thing about the threshold theorem is that the extra work needed scales reasonably, only as polynomial in the log of $\frac{T}{\varepsilon}$. Scaling up to a really really big quantum computer is a lot of work even for an ideal circuit, but adding fault-tolerance to the circuit does not require much more effort than it does to add fault-tolerance to a merely big quantum computer. Unfortunately, while the asymptotic scaling for large $\frac{T}{\varepsilon}$ is good, the actual amount of overhead needed is still pretty large since the constants hidden by the big-Oh notation are significant.

We sometimes talk about the threshold for a specific code family, and sometimes just about "the threshold" for fault tolerance. In the latter case, we are talking about the best (i.e., highest) threshold p_T optimized over all possible families of codes. In addition, different fault-tolerant protocols for the same code family can lead to substantially different "threshold" values, but the real threshold is defined using the best possible FT protocol.

The threshold theorem can be extended in various ways. One obvious way, still sticking to a basic model for fault tolerance, is to let the error rates for different types of locations be different. In that case, the threshold is no longer a single number, but a surface in the 4D space of error rates (p_P, p_G, p_S, p_M). The threshold surface separates the noiseless point $(0, 0, 0, 0)$ from the completely noisy point $(1, 1, 1, 1)$, and the extended theorem then says that we can achieve error rate ε for the outcome distribution for any point on the noiseless side of the surface with polylogarithmic overhead.

9.7 Comparatively Evaluating Fault-Tolerant Protocols

Here is a checklist to think about while deciding between FT protocols:

- There is no "best."
- Check for assumptions.

- Check which aspects of the protocol are included.
- Consider the tradeoffs:
  - Threshold,
  - Overhead,
  - Connectivity of the physical qubits,
  - Speed of syndrome decoding algorithms,
  - Behavior under different error models,
  - More.
- Compare the degree of optimization.

10 If Only It Were So Easy

Transversal Gates and Gate Teleportation

Definition. Let $\mathcal{Q}$ be a quantum operation acting on a Hilbert space $\mathcal{H}_{2^n}^{\otimes m}$, interpreted as m blocks each consisting of n qubits. Then $\mathcal{Q}$ is *transversal* if $\mathcal{Q} = \bigotimes_{i=1}^{n} \mathcal{O}_i$, where $\mathcal{O}_i$ acts on the 2^m-dimensional Hilbert space composed of the i^{th} qubit from each of the m blocks.

Transversal gates satisfy the GPP by definition, which makes them an attractive choice for fault-tolerance. They are especially useful because they conjugate pre-existing errors into errors of the same weight; i.e., they have the property that

$$\mathscr{G}_0 \circ \mathscr{F}_r = \mathscr{F}_r \circ \mathscr{G}_0 \circ \mathscr{F}_r \,.$$

Proposition. Suppose the circuit $\mathcal{Q}$ is a gate gadget for the QECC Q and $\mathcal{Q}$ is transversal. Then $\mathcal{Q}$ satisfies the GPP and the GCP.

Proposition. The product of transversal gate gadgets is a transversal gate gadget.

Theorem. Clifford group circuit U maps codewords of a stabilizer S to codewords of the same stabilizer S if and only if $UMU^\dagger \in S$ for all $M \in S$.

Corollarly. Clifford group circuit U maps codewords of a stabilizer S to codewords of the same stabilizer S if and only if $UMU^\dagger \in S$ for all generators M of S.

$\star$ Since $\mathsf{N}(S)/S$ is the logical Pauli group for a stabilizer S, the transformation performed on $\mathsf{N}(S)/S$ by a Clifford gate will tell us what Clifford operation has been performed on the encoded state.

$\star$ We can implement the whole logical Clifford group on m blocks of the Steane code just by doing transversal operations. The transversal U does not always give us the logical U—instead it gives us the complex conjugate U^*—but nevertheless, it is straightforward to perform any logical Clifford group operation for this code.

Theorem. A stabilizer code $\mathsf{Q}(S)$ has a gadget consisting of transversal CNOT if and only if $\mathsf{Q}(S)$ is a Calderbank-Shor-Steane code. If so, S has a choice of logical operators for which the gadget simulates a location consisting of the tensor product of CNOTs from logical qubit i in the first block to logical qubit i in the second block, for all i.

Theorem. For any Calderbank-Shor-Steane code, transversal measurement followed by classical decoding is a fault-tolerant gadget implementing logical measurement of all encoded qubits.

Proof sketch. The theorem is true because the codewords are superpositions of states from a classical error-correcting code. That means a small number of errors in the classical output cannot confuse us between logical codewords.

Theorem. *Eastin-Knill.* If Q is an $((n, K, d))_q$ QECC with $d \geq 2$, and G is the set of logical unitary gates that have transversal gadgets for Q, then G is a discrete group. In particular, G is not dense in $\mathsf{SU}(K)$.

$\star$ The Eastin-Knill theorem tells us that transversal gate gadgets can never be enough to get a universal set of gates.

10.1 Gate Teleportation

Now we'll get started on figuring out how to get a universal set of gates. To do so, assume we have the ability to perform Clifford group gates and Pauli measurements, and the ability to prepare an arbitrary state of our choice. One thing we can do with the available tools is teleport a quantum state; this requires a Bell measurement, an EPR pair, and a Pauli operator (with classical control). Suppose Alice wants to perform a gate U (which might be a Clifford group gate or a non-Clifford group gate) on a state $|\psi\rangle$. She can take the rather perverse route of teleporting state $|\psi\rangle$ to Bob by doing a Bell measurement and sending over two classical bits, and having Bob perform U on $|\psi\rangle$. However, in this scenario, Bob can be impatient and perform U on a fixed state (EPR pair) $|00\rangle + |11\rangle$ before he's heard from Alice. But when he does get the classical bits from Alice, he should undo U, do the appropriate Pauli (dictated by the classical bits sent by Alice), and then redo U. In other words, if Alice's message indicates that he is supposed to perform P, he should instead do $UPU^\dagger$. This is actually progress: Bob's impatience means that he performs the gate U on a fixed state $|00\rangle + |11\rangle$, regardless of Alice's state $|\psi\rangle$. That means we've converted the task of performing U into three steps:

Protocol. *Gate Teleportation.*

1) Create $(I \otimes U)(|00\rangle + |11\rangle)$ shared between Alice and Bob.

2) Alice performs a Bell measurement and sends the results to Bob.

3) Bob performs $UPU^\dagger$ when Alice's classical bits correspond to Pauli P.

For general U, the gate $UPU^\dagger$ could be a pretty complicated thing. There's no reason to assume it will be any easier to do than U. But for certain special U, the conjugated Pauli always is simpler than U. For instance, if $U \in \mathsf{C}_1$, then $UPU^\dagger \in \mathsf{P}_1$, and Bob only needs to

perform a Pauli at step 3. It turns out that there are a number of gates U with the property that whenever $P \in \mathsf{P}_n$, then $UPU^\dagger \in \mathsf{C}_n$.

⋆ This gate teleportation construction therefore is a way to bootstrap from simpler operations to more complicated ones.

If Alice and Bob can request whatever ancilla states they need, then they can use the gate teleportation construction to build up more and more complicated gates. Given Pauli gates, they can perform Clifford gates, and given Clifford gates, they can perform any gate U with the property $P \in \mathsf{P}_n \implies UPU^\dagger \in \mathsf{C}_n$.

⋆ While the gate teleportation protocol is described explicitly for a single-qubit gate U, that is not at all necessary. An n-qubit gate can be done in the same way just using the state $(I \otimes U)(|00\rangle + |11\rangle)^{\otimes n}$ instead of the 1-qubit version. In step 2 Alice teleports all her qubits, and in step 3, the Paulis run over n-qubit Paulis.

⋆ The state $(I \otimes U)(|00\rangle + |11\rangle)$ is known as a *magic state*. The gate teleportation protocol is an example of a class of protocols called *magic state injection*, whereby a magic state is used to perform a non-Clifford group unitary on some data using only Clifford group gates and Pauli measurements. The term "magic state" is used somewhat loosely to mean one of three things:

1. It could be a state like this that can be used directly for magic state injection.
2. It could be a state (pure or mixed) that can be further processed by Clifford group operations, at which point it could then be used to perform a non-Clifford group gate.
3. It could refer to any non-stabilizer pure state.

10.1.1 $\mathcal{C}_k$ Gates

Definition. Let $\mathcal{C}_1 = \mathsf{P}_1$ and let

$$\mathcal{C}_k = \{U \in \mathsf{U}(2^n) : UPU^\dagger \in \mathcal{C}_{k-1} \ \forall P \in \mathsf{P}_n\}\,.$$

The collection of the sets $\{\mathcal{C}_k\}$ is known as the *Clifford hierarchy*. The number of qubits n is implicit in this notation, but $\mathcal{C}_k$ can also refer to the union of $\mathcal{C}_k$ over all values of n.

⋆ If we can perform arbitrary $\mathcal{C}_k$ gates, then we can use gate teleportation to directly perform $\mathcal{C}_{k+1}$ gates, which then lets us perform $\mathcal{C}_{k+2}$ gates, and so on.

⋆ $\mathcal{C}_k$ is not closed under multiplication for $k \geq 3$.

10.1.1.1 Examples

1. The first example is the $T = \sqrt[4]{Z} = R_{\frac{\pi}{8}}$ gate:

$$TXT^\dagger = XS^\dagger$$
$$TZT^\dagger = Z\,,$$

   where $S = \sqrt{Z} = R_{\frac{\pi}{4}} \in \mathsf{C}_1 = \mathcal{C}_2$ is the phase gate.

2. Another gate in $\mathcal{C}_3$ is the Toffoli gate Tof:

$$\begin{aligned}
\mathrm{Tof}(X \otimes I \otimes I)\,\mathrm{Tof} &= X_1\mathrm{CNOT}_{2,3} \\
\mathrm{Tof}(I \otimes X \otimes I)\,\mathrm{Tof} &= \mathrm{CNOT}_{1,3}X_2 \\
\mathrm{Tof}(I \otimes I \otimes X)\,\mathrm{Tof} &= X_3 \\
\mathrm{Tof}(Z \otimes I \otimes I)\,\mathrm{Tof} &= Z_1 \\
\mathrm{Tof}(I \otimes Z \otimes I)\,\mathrm{Tof} &= Z_2 \\
\mathrm{Tof}(I \otimes I \otimes Z)\,\mathrm{Tof} &= \mathrm{CZ}_{1,2}Z_3\,.
\end{aligned}$$

This is why the universal gate sets $\langle \mathsf{C}_n, T\rangle$ and $\langle \mathsf{C}_n, \mathrm{Tof}\rangle$ are particularly interesting for fault tolerance: they each consist of the Clifford group plus a single $\mathcal{C}_3$ gate, which can be implemented via gate teleportation.

$\star$ For fixed k and n, $\mathcal{C}_k/\{e^{\iota\theta}I\}$ (which is $\mathcal{C}_k$ with global phases removed) must be finite. Hence, $\mathcal{C}_k$ is not dense in $\mathsf{U}(2^n)$ even for large k. This means that direct constructions via teleportation are not enough. In order to get a universal set of gates, we can construct a $\mathcal{C}_3$ gate such as T via teleportation, but then we will need to multiply gates together to get good approximations of arbitrary unitary transformations.

10.1.2 Universal Fault Tolerance

Theorem. For any Clifford group gate $U \in \mathsf{C}_n$ and any stabilizer code, there exists a fault-tolerant gate gadget for U.

Theorem. There exists a fault-tolerant protocol for any stabilizer code.

10.1.3 Compressed Gate Teleportation

In our gate teleportation protocol, even for a single-qubit gate, the construction demands a two-qubit ancilla, and an n-qubit gate needs a $2n$-qubit ancilla. For certain gates, we can reduce the size of the construction and use a smaller ancilla using one-bit teleportation circuits.

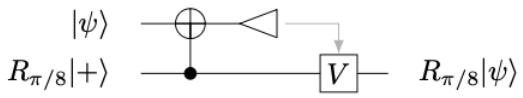

Figure 7: A smaller circuit for gate teleportation with the T gate. Here, $V = TXT^\dagger$.

11 Who Corrects The Correctors?

Fault-Tolerant Error Correction And Measurement

11.1 Fault-Tolerant Pauli Measurement

Suppose we want to non-destructively measure a Hermitian Pauli $P = P_1 \otimes \cdots \otimes P_m$ on a code block, omitting the qubits on which P is the identity. The obvious procedure is to prepare an ancilla in $|+\rangle$, perform controlled-P with the ancilla as control, and measure the ancilla in the X basis. If the data is a $(-1)^b$ eigenstate of P, the ancilla becomes $|0\rangle + (-1)^b|1\rangle$, so the measurement gives b. For a general input, the procedure projects the data with $\frac{1}{2}(I + (-1)^b P)$.

Unfortunately, a bit flip on the ancilla halfway through the controlled-P propagates to every data qubit it subsequently interacts with. One fault can therefore cause many errors in a single block. We need to distribute the control over several ancilla qubits.

Definition. An m-qubit *cat state* is the state

$$|\mathrm{cat}_m\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes m} + |1\rangle^{\otimes m}) \,.$$

It is also known as a *GHZ state*.

Protocol. *Single-copy cat state measurement.*

1) Prepare and verify an m-qubit cat state, where $m = \mathrm{wt}\, P$.
2) Perform controlled-P_i from the i^{th} cat state qubit to the corresponding data qubit.
3) Measure every cat state qubit in the X basis. The product of the outcomes is the measured eigenvalue of P.

Equivalently, apply $H^{\otimes m}$ to the cat state and measure in the computational basis. Ignoring normalization,

$$H^{\otimes m}(|0\rangle^{\otimes m} + (-1)^b|1\rangle^{\otimes m}) = \sum_x [1 + (-1)^{b+\mathrm{wt}\, x}]|x\rangle \,.$$

Thus, the parity of the measured bit string is b.

11.1.1 Cat State Verification

Creating the cat state by applying H to one qubit and then a sequence of CNOTs is not fault-tolerant. A single bit flip during the preparation can produce several bit flips in the cat state, which then propagate into several data errors. We must check the cat state before it touches the data.

The cat state is stabilized by $X^{\otimes m}$ and $Z_i Z_{i+1}$ for $i = 1, \ldots, m-1$. To check whether two cat qubits agree, prepare a check qubit in $|0\rangle$, apply CNOTs from the two cat qubits into it, and measure it in the computational basis. This measures their parity without learning their individual values and hence without destroying the cat state superposition. A failed check means we discard the cat state and try again, or perform additional checks and correct it. The checks must be chosen and repeated to catch dangerous errors from up to t faults, including faults in the checks themselves.

$\star$ The orientation of these CNOTs matters. Bit flips can propagate from the cat state into the check qubit, but cannot propagate back from that target into a second cat qubit. Phase

errors can propagate back, but all single-qubit Z errors have the same action on the cat state. An even number of Z errors acts trivially, and an odd number is equivalent to one Z error. This degeneracy is what makes the verification work.

Let $\mathscr{C}_s$ denote a single-copy cat state measurement with s faults, including faults in the verified preparation. Then, for $r + s \leq t$,

$$\mathscr{C}_s \circ \mathscr{F}_r = \mathscr{F}_{r+s} \circ \mathscr{C}_s \circ \mathscr{F}_r \,.$$

Here the filter on the output acts on the surviving data block. Each fault adds at most one data error. However, a single phase error in the cat state or a single flipped measurement outcome can still give the wrong classical bit.

11.1.2 Repeated Measurement

Repeating the cat state measurement alone does not give a fault-tolerant logical measurement. If the input is $E\big|\overline{\psi}\big\rangle$ and $\{E, P\} = 0$, every faultless repetition measures the opposite eigenvalue from that of the decoded state. Furthermore, a fault in one repetition can change the data syndrome and spoil all later repetitions. We therefore intersperse the measurements with FTEC steps.

Theorem. For a code correcting t errors, $2t + 1$ single-copy cat state measurements of a logical Pauli, with FTEC between consecutive repetitions and majority vote on the outcomes, satisfy the MCP.

Proof sketch. Let the input pass an r-filter and let the whole gadget have s faults, with $r + s \leq t$. A measurement preceded by a faultless FTEC and itself having no faults gives the correct logical outcome. For the first measurement, an input error takes the place of a fault in the preceding FTEC. Charge each possibly wrong outcome to an input error, a fault in its measurement, or a fault in its preceding FTEC. There are at most $r + s \leq t$ such outcomes. The remaining at least $t+1$ outcomes agree. The error propagation property and ECP ensure that, apart from the intended projection, the logical state is preserved throughout.

$\star$ We do not use the full repeated logical measurement gadget to measure each stabilizer generator during error correction. That would require error correction inside error correction. Instead, we use the single-copy cat state measurement as a component and repeat the full syndrome extraction.

11.2 Shor Error Correction

A single pass through the stabilizer generators is not sufficient. There are two different things that can go wrong:

1) An ancilla or measurement fault can give an incorrect syndrome bit.

2) A data error can occur between measurements of different generators. Each bit may be correct at the time it is measured, but the assembled syndrome may never have been the syndrome of the data at any one time.

For example, for the five-qubit code, a Z_2 error occurring after the first two syndrome bits have been measured can produce the hybrid syndrome 0001, which is the syndrome of X_1. Applying the corresponding correction then leaves two errors from one fault.

Protocol. *Shor error correction.* Let the code correct t errors.

1) Measure every stabilizer generator using single-copy cat state measurement with a fresh verified cat state for each generator. This gives one full syndrome.

2) Repeat the full syndrome extraction $(t+1)^2$ times.

3) Find the last run of $t+1$ consecutive identical syndromes and use that syndrome. If there is no such run, use the longest run, choosing the latest in case of a tie.

4) Choose a Pauli correction with that syndrome, of weight at most t whenever such a correction exists, and apply it. Assign a correction to every syndrome, including those with no representative of weight at most t.

Proposition. If there are at most t faults, $(t+1)^2$ syndrome extractions suffice to obtain $t+1$ consecutive identical syndromes.

Proof sketch. There are at most t faulty extractions. If there were no run of $t+1$ faultless extractions, the at most $t+1$ runs of faultless extractions would each have length at most t. Together with the faulty extractions, this gives at most $t(t+1)+t = (t+1)^2 - 1$ extractions. A faultless run measures a fixed syndrome throughout.

$\star$ We repeat the *whole syndrome*, not each bit separately. Nor can we simply take a majority of $2t+1$ full syndromes: there are many possible syndromes, and faults can change the true syndrome during the gadget, so there need not be a majority.

Theorem. Shor error correction satisfies the ERP and ECP.

Proof sketch.

1) Among the $t+1$ identical syndrome extractions used by the protocol, at least one is faultless. Let E be the data error at that extraction. The measured syndrome is therefore the syndrome of E.

2) Write the subsequent data error as F, the chosen correction as E', and faults in applying the correction as G. The output is

$$GE'FE\left|\overline{\psi}\right\rangle = \pm GF(E'E)\left|\overline{\psi}\right\rangle .$$

3) Since E' and E have the same syndrome, $E'E \in \mathsf{N}(S)$, up to phase. Thus, $(E'E)\left|\overline{\psi}\right\rangle$ is a codeword. Also, $\mathrm{wt}(GF) \leq s$, where s is the total number of gadget faults. This proves the ERP, without any bound on the input error.

4) If the input has at most r errors and $r+s \leq t$, then $\mathrm{wt}\, E \leq t$. The decoder chooses E' in the same correctable class, so $E'E$ acts trivially on the code. Ideal decoding removes GF and recovers the original logical state. This proves the ECP.

⋆ The ERP says that the output is close to *some* codeword. It does not say that this is the right codeword when the input has too many errors. The ECP is the additional condition that preserves the encoded information.

11.3 Steane Error Correction and Measurement

Shor EC uses simple ancillas but many interactions with the data. We can instead put more of the work into ancilla preparation, where mistakes can be detected before they affect the unknown logical state. Steane EC does this for CSS codes using encoded $|0\rangle$ and $|+\rangle$ ancillas.

Protocol. *Steane error correction.*

1) To extract the bit flip syndrome, prepare a verified $|\overline{+}\rangle$ block and perform transversal CNOT from the data into the ancilla. Measure the ancilla transversally in the Z basis and classically calculate its error syndrome.

2) To extract the phase error syndrome, prepare a verified $|\overline{0}\rangle$ block and perform transversal CNOT from the ancilla into the data. Measure the ancilla transversally in the X basis and classically calculate its error syndrome.

3) Use the two syndromes to choose a Pauli correction on the data.

For a code encoding several qubits, the ancillas have all logical qubits in the indicated state. The two syndrome extraction steps can also be performed in the reverse order.

The logical CNOT does nothing to $|\overline{\psi}\rangle|\overline{+}\rangle$, since $X|+\rangle = |+\rangle$. Nevertheless, physical bit flips on the data propagate into the ancilla. Measuring the ancilla gives a randomly chosen classical codeword with those bit flips. Its syndrome reveals the errors, while its logical value is random and gives no information about the data. Similarly, a $|\overline{0}\rangle$ control does not change the logical target, but phase errors propagate backwards from the data into the ancilla.

If e is the bit flip pattern on the data, f the pattern already on the ancilla, and g the additional pattern from measurement faults, then the measured classical syndrome is that of $e + f + g$. Meanwhile, phase errors on that ancilla can propagate back into the data. Thus, both types of ancilla error matter: one contaminates the inferred syndrome and the other contaminates the data directly.

11.3.1 Steane Measurement

Protocol. *Non-destructive Steane measurement of logical Z.* Prepare a verified $|\overline{0}\rangle$ ancilla, perform transversal CNOT from the data into the ancilla, and measure the ancilla transversally in the Z basis. Classically decode the measured word; its logical value is the measurement outcome. Logical X measurement uses the analogous construction with a $|\overline{+}\rangle$ ancilla, reversed CNOT, and X measurement.

⋆ The difference between syndrome extraction and logical measurement is the ancilla state. A $|\overline{+}\rangle$ target hides the logical value but reveals the bit flip syndrome. A $|\overline{0}\rangle$ target reveals both. This is why syndrome extraction does not accidentally measure the encoded information.

Definition. A non-destructive measurement gadget satisfies the *measurement propagation property (MPP)* if, whenever $r + s \leq t$,

$$\mathscr{M}_s \circ \mathscr{F}_r = \mathscr{F}_{r+s} \circ \mathscr{M}_s \circ \mathscr{F}_r \,.$$

Its correctness property also retains the quantum output: ideal decoding after the gadget gives the same quantum state and classical outcome as ideal non-destructive measurement after decoding the input.

Theorem. With fault-tolerant ancilla preparation, Steane EC satisfies the ERP and ECP. Non-destructive Steane measurement satisfies the MPP and the non-destructive measurement correctness property.

$\star$ No repetition of the syndrome extraction is required for these properties. A wrong syndrome caused by a small ancilla error leads to a correspondingly small residual data error. In Shor EC, changing one syndrome bit can suggest a completely different error; in Steane EC, a single faulty measurement changes one bit of a classical codeword, whose redundancy is still available to the decoder. Further repetition may improve performance but is not needed to establish fault tolerance.

11.4 Knill Error Correction and Measurement

Knill EC uses teleportation to move the logical state into a fresh block while extracting the information needed to correct errors. It works for any stabilizer code, not only CSS codes.

Protocol. *Knill error correction.*

1) Fault-tolerantly prepare an encoded Bell pair, with one Bell pair for each logical qubit in a block.
2) Perform transversal CNOT from the data block into the first half of the encoded Bell pair.
3) Measure every data qubit in the X basis and every qubit in the first ancilla block in the Z basis.
4) Classically decode the physical Bell measurement outcomes to determine the logical Bell measurement outcome.
5) Apply the corresponding logical Pauli to the second ancilla block. This block is the output.

Why does this work for a non-CSS code, even though transversal CNOT need not be a logical gate? The CNOT and the two measurements together measure $X_i \otimes X_i$ and $Z_i \otimes Z_i$ for each physical position i. These commute. By multiplying their outcomes, with the appropriate Pauli phases, we can determine the eigenvalues of $M \otimes M$ for every stabilizer generator M, and of $\overline{P} \otimes \overline{P}$ for logical Paulis $\overline{P}$.

If the data and measured ancilla blocks have Pauli errors E and F, the stabilizer parities reveal the syndrome of EF. When this error is correctable, we can account for its effect on

the logical parities and recover the correct logical Bell outcome. The individual syndromes of E and F need not be separated.

⋆ The transversal CNOT by itself does not have to preserve the code. It is the *whole Bell measurement procedure* that performs the required logical operation.

Protocol. *Knill measurement of logical Z.* Replace the Bell-pair ancilla with a verified $|\overline{0}\rangle$ block and perform the same physical Bell measurement and classical decoding. Keep the logical $\overline{Z} \otimes \overline{Z}$ outcomes. Since the ancilla has logical $Z = +1$, these are the desired data outcomes. Discard the logical $X \otimes X$ outcomes.

Theorem. Knill EC satisfies the ERP and ECP, and Knill measurement satisfies the MCP.

Proof sketch. For correctness, when $r + s \leq t$, the combined data, ancilla, and measurement errors are within the decoding capability, so the inferred logical Bell outcome is correct. Teleportation therefore transfers the right logical state. For recovery, the output block is a fresh ancilla block. Its physical errors come only from its preparation, storage, and final transversal Pauli correction, and have weight at most s regardless of the input error.

11.5 Comparing FTEC Protocols

- *Shor EC:* Any stabilizer code; verified cat states; many interactions with the data and repeated full syndrome extraction.
- *Steane EC:* CSS codes; verified encoded $|0\rangle$ and $|+\rangle$ states; two transversal interactions with the data, followed by ancilla measurements.
- *Knill EC:* Any stabilizer code; a verified encoded Bell pair; one transversal interaction with the old data, followed by physical Bell measurement. The output occupies a fresh block.

Shor EC is appealing when only small ancillas are available or the stabilizer generators have low weight. For large-weight generators, phase errors make large cat states unreliable: if the phase error probability per cat qubit is p, the probability of no phase errors is $(1-p)^m$. Once mp is appreciable, many repetitions may be needed. Steane and Knill ancillas instead have protection against both bit and phase errors.

Definition. A *low-density parity check (LDPC) code* is a code with a parity check description in which every check involves only a bounded number of qubits and every qubit participates in only a bounded number of checks, independent of the block size.

⋆ Small check weight, rather than small total block size, is what keeps the cat states small. This is one reason LDPC codes are significant for fault tolerance.

Knill EC also combines naturally with gate teleportation: use an appropriate entangled ancilla and perform a gate while correcting errors. Replacing the physical data qubits is useful for leakage and for moving away from particularly noisy qubits.

11.5.1 The Pauli Frame

Definition. The *Pauli frame* is the classical record of known Pauli corrections that have not been physically applied. There is a separate frame for each code block.

Suppose the state has a known Pauli E relative to the desired state. Instead of correcting E, we can keep track of it. The state occupies the equally good code with stabilizer $ESE^\dagger$, which differs from S only in generator signs. If a Clifford gate U is applied, update the frame by

$$E \longmapsto UEU^\dagger .$$

If a Pauli P is measured, reverse the interpretation of its outcome precisely when $\{E, P\} = 0$.

New syndrome information is interpreted relative to the current frame. The accumulated frame may eventually have high weight or be a logical Pauli; that is fine because it is known. What must remain small is the unknown error relative to the frame.

⋆ A Pauli frame is a classical record, not an assumption that the quantum state is error-free. Non-Clifford gates require special care because $UEU^\dagger$ need not be a Pauli. The implementation or its measurement-dependent corrections must account for the current frame.

12 Any Sufficiently Advanced Fault-Tolerant Protocol Is Indistinguishable From Magic States

State Preparation And Its Applications

12.1 Preparation of Encoded Stabilizer States

To prepare a particular encoded stabilizer state, we must check both the stabilizer of the QECC and the logical stabilizers specifying the state. For an $[[n, k, d]]$ code, these give $n - k$ code generators and k independent logical generators, for a total of n independent stabilizers.

Protocol. *State preparation using Shor EC.*

1) Prepare the desired encoded stabilizer state using a Clifford encoding circuit. This circuit need not be fault-tolerant.

2) Perform Shor EC for the full stabilizer of the state, including the logical generators.

3) Apply the Pauli correction indicated by the syndrome.

Theorem. State preparation using Shor EC verification satisfies the PPP and PCP.

Proof sketch. The ERP for the full stabilizer ensures that the output differs from its unique codeword by at most s errors when the verification has s faults, regardless of the state produced by the initial encoder. This gives the PPP for the underlying QECC. Since $s \leq t$, ideal decoding removes those errors and gives the desired logical state, proving the PCP.

⋆ Checking only the code stabilizers is insufficient. The wrong logical state can have trivial error syndrome. We must also check the logical operators whose eigenvalues specify the state being prepared.

There are two useful variations:

1) Skip the initial encoder and start from an arbitrary state. Measuring all n independent generators projects onto a one-dimensional joint eigenspace. The measured syndrome identifies that state and hence the Pauli needed to turn it into the desired state.

2) Use error detection instead of correction. After the initial encoder, repeat the full syndrome extraction t times, rejecting whenever any syndrome is nonzero. With at most t faults, an error from the initial encoder cannot be hidden by faults in all t checks. Faults late in verification can leave residual errors, but no more than the number of faults.

If we combine the variations, we should instead require $t + 1$ identical full syndromes and then correct the accepted state. The syndrome need not be zero. At least one extraction is faultless, so the accepted value was correct at some point.

12.1.1 Preparation of CSS Codewords

Encoded CSS ancillas can also be checked using transversal CNOTs and measurements, comparing independently prepared copies of the desired state. However, the ancillas used for comparison must themselves be checked. Otherwise, a correlated error from one faulty encoder can propagate into the state we intend to keep.

One general construction has two levels of verification. Prepare groups of $t + 1$ copies and compare them for phase errors, discarding groups that fail. Then compare $t+1$ surviving states for bit flip errors and retain one state if the checks pass. This uses $(t+1)^2$ initial preparations per attempt. The order of the checks is chosen so that dangerous errors propagated in the first stage are detected in the second. More economical constructions are possible for particular codes and encoding circuits.

⋆ Post-selection trades extra preparations for more reliable ancillas. Unlike the unknown data, a rejected ancilla can simply be discarded. Preparing several candidates in parallel can keep the data from waiting for a successful attempt.

12.2 Clifford Eigenstate Preparation

Gate teleportation converts the problem of implementing certain non-Clifford gates into the problem of preparing their magic states. We still need a fault-tolerant way to make those states.

Proposition. Let $|\psi\rangle = U|\phi\rangle$, where $U \in \mathcal{C}_k$ and $|\phi\rangle$ is a stabilizer state with stabilizer S. Then $|\psi\rangle$ is the unique common $+1$ eigenstate of $UMU^\dagger$ for $M \in S$, and each such operator belongs to $\mathcal{C}_{k-1}$.

Proof. If $M|\phi\rangle = |\phi\rangle$, then

$$UMU^\dagger|\psi\rangle = UM|\phi\rangle = U|\phi\rangle = |\psi\rangle\,.$$

Conjugating the rank-one stabilizer projector gives

$$\Pi_\psi = U\Pi_\phi U^\dagger = \frac{1}{2^n}\prod_{i=1}^{n}(I + UM_iU^\dagger)\,,$$

where M_i generate S. Thus, the common eigenspace is one-dimensional. Membership in $\mathcal{C}_{k-1}$ follows from the definition of the Clifford hierarchy.

Lemma. Suppose a state is uniquely specified by its eigenvalues for a set of Clifford operators. For any code with a fault-tolerant implementation of the full Clifford group, there is a fault-tolerant preparation gadget for that state. If the operators have the form $VP_iV^\dagger$, where the P_i generate a stabilizer state and $V \in \mathcal{C}_3$, the preparation can be done without post-selection.

The idea is to generalize cat state measurement from Paulis to Clifford operators. If a logical Clifford has a transversal implementation, let each cat qubit control one factor of that implementation. For eigenvalues ± 1, the same parity measurement works. For a finite-order operator U, with $U^r = I$, use an r-dimensional cat state, controlled powers of the gates, and an inverse Fourier transform. The sum of the measured qudit values modulo r identifies the eigenvalue. Verification, repetition, and interspersed error correction are still required.

For non-transversal Clifford gadgets, one can use a separate cat state control for each gate in a coherent implementation of the gadget. A fault in a verified cat state then affects only one gate of the underlying fault-tolerant circuit.

Protocol. *Preparation of $V|\phi\rangle$, with $V \in \mathcal{C}_3$.*

1) Prepare an encoded starting state.

2) Fault-tolerantly measure the logical operators $VP_iV^\dagger$, where the P_i generate the stabilizer of $|\phi\rangle$.

3) Let s record the incorrect eigenvalues. Choose a Pauli $Q(s)$ that changes those eigenvalues to those of $|\phi\rangle$ in the original stabilizer basis.

4) Apply $VQ(s)V^\dagger$, which is a Clifford gate, fault-tolerantly.

⋆ The controlled gates needed for these measurements are physical operations from the available universal gate set. We are not assuming that controlled-Clifford gates are themselves Clifford gates.

12.3 Magic State Distillation

Definition. *Magic state distillation* is a procedure that consumes several imperfect magic states and, using Clifford gates and Pauli measurements, produces fewer magic states with higher fidelity. It generally includes post-selection on measurement outcomes.

For analysis, we first assume Clifford operations and Pauli measurements are perfect, while the supplied magic states are noisy. In an actual computation, the whole procedure is performed on encoded qubits whose Clifford operations have already been made reliable. The code used to protect those operations need not be the code used by the distillation protocol.

12.3.1 Distillation Using the 15-Qubit Code

Let $|A\rangle = T|+\rangle$. The $[[15, 1, 3]]$ code has a transversal T implementing logical $T^\dagger$. Since its transversal $S^\dagger$ implements logical S, the combination performs logical T.

Protocol. *15-to-1 magic state distillation.*

1) Begin with 15 noisy $|A\rangle$ states.
2) Prepare $|\overline{+}\rangle$ in the 15-qubit code using Clifford operations.
3) Use the 15 magic states in compressed gate teleportation to implement physical T on each of the 15 code qubits.
4) Apply transversal $S^\dagger$.
5) Measure the code syndrome. If it is nonzero, discard the block and restart.
6) Decode the accepted block using Clifford operations, keeping the logical qubit.

Proposition. Under independent input errors of probability p, and perfect Clifford operations and measurements, the accepted output has error probability $O(p^3)$ for sufficiently small p.

Proof sketch. Each faulty input magic state affects only one qubit of the 15-qubit code. The code detects errors on one or two qubits, so an undetected error that changes the logical state requires at least three faulty inputs. Such events have probability $O(p^3)$, while the acceptance probability approaches 1 as $p \to 0$.

⋆ Detection is preferable to correction here. A distance-three code can correct one arbitrary error but detect two. Since this is state preparation, we can discard a detected error instead of trying to repair it.

A compact version measures the 15-qubit code syndrome directly on 15 noisy $|A\rangle$ states, rejects a nonzero syndrome, decodes, and applies S to the decoded qubit. It has the same cubic suppression mechanism, although the acceptance behavior need not be the same as in the encoded-$|+\rangle$ construction.

If one round gives $p' \le cp^3$, repeated rounds satisfy

$$p_j \le c^{-1/2}(c^{1/2}p)^{3^j},$$

provided the input is in the regime where this bound contracts. Two rounds use up to 15^2 inputs per final attempt and give $O(p^9)$ error; j rounds give $O(p^{3^j})$, before accounting for errors in the Clifford circuitry.

⋆ Distillation does not remove errors introduced by its own imperfect logical gates. In practice, the accuracy of those gates must improve along with the target magic-state accuracy. The ideal-Clifford model isolates the improvement due to distillation itself.

12.3.2 Twirling Magic States

Theorem. Let $\{|\psi_i\rangle\}$ be an orthonormal basis and let U_a be finite-order Clifford operators with

$$U_a|\psi_i\rangle = \lambda_{ai}|\psi_i\rangle\,.$$

Suppose the vectors of eigenvalues $(\lambda_{ai})_a$ distinguish all basis states. If $U_a^{m_a} = I$, applying $U_a^{r_a}$ for each a, with r_a uniformly random in $\{0,\dots,m_a-1\}$, transforms any state ρ into

$$\rho' = \sum_i \langle\psi_i|\rho|\psi_i\rangle|\psi_i\rangle\langle\psi_i|\,.$$

Proof sketch. In this basis, averaging over powers of U_a multiplies the ij matrix entry by

$$\frac{1}{m_a}\sum_{r=0}^{m_a-1}(\lambda_{ai}\lambda_{aj}^*)^r\,.$$

This is 1 if the two eigenvalues agree and 0 otherwise. Every off-diagonal entry is eliminated by at least one of the averages, while diagonal entries are unchanged.

Definition. *Twirling* is averaging over a group of operations to simplify a state or noise model.

For $|A\rangle = T|+\rangle$, use the Hermitian Clifford $V = TXT^\dagger = (X+Y)/\sqrt{2}$. Applying I or V with probability $\frac{1}{2}$ produces

$$\rho' = (1-p)|A\rangle\langle A| + pZ|A\rangle\langle A|Z\,,$$

where $1-p = \langle A|\rho|A\rangle$. Thus, a general single-state error can be replaced by a stochastic error model without changing the fidelity to the target.

$\star$ Twirling does not itself purify the state, and independent local twirls do not make initially correlated input states independent.

12.3.3 Distillation Using the Five-Qubit Code

Not all distillation procedures come from transversal non-Clifford gates. Let R be the Clifford that conjugates $X \mapsto Z \mapsto Y \mapsto X$, and let

$$|R_+\rangle = \cos\theta|0\rangle + e^{\iota\pi/4}\sin\theta|1\rangle, \qquad \cos(2\theta) = \frac{1}{\sqrt{3}}\,.$$

The orthogonal state $|R_-\rangle$ lies at the opposite point of the Bloch sphere. Their projectors are

$$|R_\pm\rangle\langle R_\pm| = \frac{1}{2}\left[I \pm \frac{X+Y+Z}{\sqrt{3}}\right].$$

Protocol. *Five-qubit magic state distillation.*

1) Begin with five noisy $|R_+\rangle$ states.

2) Twirl each in the R eigenbasis by applying I, R, or R^2 with equal probability.

3) Measure the syndrome of the five-qubit code and reject unless it is zero.

4) Decode, keep the logical qubit, and apply YH.

With five perfect inputs, the acceptance probability is $\frac{1}{6}$. The accepted decoded state before YH is $|R_-\rangle$; the final gate maps it back to $|R_+\rangle$. An input with exactly one $|R_-\rangle$ factor has zero projection onto the code, whereas some inputs with two such factors survive. Thus, independent input error probability p gives accepted output error $O(p^2)$.

⋆ This protocol uses the transversal Clifford symmetry of the five-qubit code. Its quadratic improvement is not simply the statement that a distance-three code corrects one error: even the perfect product of magic states is only probabilistically projected onto the code.

13 If It's Worth Doing, It's Worth Overdoing

The Threshold Theorem

13.1 Adversarial Errors

Independent physical faults need not produce independent logical faults. Adjacent gadgets share error correction steps, and the logical effect of a bad gadget can depend on the incoming syndrome. To iterate a fault-tolerant simulation, we therefore need a noise model that still applies after replacing physical gadgets with logical locations.

Definition. An *adversarial local stochastic error model* is a mixture over fault paths Ω, with probabilities p_Ω, such that for every set S of locations,

$$\sum_{\Omega \supseteq S} p_\Omega \leq \prod_{L \in S} p_L \,.$$

Locations outside Ω act ideally. Locations in Ω may interact arbitrarily with a persistent environment, including correlations between different faulty locations. The environment starts in a fixed state. If every $p_L = p$, the bound is $p^{|S|}$.

⋆ The bound concerns the probability that *all locations in S are faulty*, regardless of what happens elsewhere. It is not the probability that S is the exact fault path. Also, "local" refers to the bounded number of qubits in a location, not geometric proximity.

The parameter p_L is an upper bound, not necessarily the actual marginal probability of a fault at L. For instance, an error source producing correlated pairs with probability q may require a parameter of order $\sqrt{q}$ rather than q. This allows some correlated noise, but can make the resulting threshold bound much more conservative.

⋆ A small constant probability of a failure affecting the entire computer does not obey a useful size-independent bound of this form. Nor does a small systematic overrotation at every location naturally fit a low-rate stochastic model. The latter can still be handled by a different threshold proof.

13.2 Good and Bad Extended Rectangles

Definition. An *extended rectangle (exRec)* consists of a gate or wait gadget together with all FTECs immediately before and after it. For preparation, there are only trailing FTECs; for measurement, only leading FTECs. A *rectangle (Rec)* omits the leading FTECs. A *truncated exRec* has one or more trailing FTECs removed.

Definition. For a fixed fault path and a code correcting t errors, an exRec is *good* if it contains at most t faults, and *bad* if it contains at least $t+1$. The same definitions apply to truncated exRecs, counting only the locations that remain.

⋆ Rectangles partition the simulation, but extended rectangles overlap. A trailing FTEC of one exRec is a leading FTEC of the next. Removing a shared FTEC can turn a bad exRec into a good truncated exRec.

13.2.1 Correctness

Definition. A gate rectangle $\mathscr{R}_L$ simulating a location L is *correct* if ideal decoding can be moved to its input:

$$\mathscr{D}^{\otimes m} \circ \mathscr{R}_L = L \circ \mathscr{D}^{\otimes m},$$

on the states entering the rectangle in the simulation, where m is the number of blocks. For preparation, correctness is $\mathscr{D} \circ \mathscr{R}_L = \mathcal{P}$; for measurement, it is $\mathscr{R}_L = \mathcal{M} \circ \mathscr{D}$. An exRec is correct when its rectangle is correct. Truncated rectangles have the analogous condition on the outputs that remain.

Theorem. A good exRec is correct. A good truncated exRec is also correct.

Proof sketch. The leading FTECs reduce input errors to at most the faults in those FTECs, by the ERP. Since the entire exRec has at most t faults, the gate input errors and gadget faults satisfy the hypotheses of the GPP and GCP. The trailing FTECs satisfy the ECP. Applying these identities moves ideal decoders left through the rectangle, leaving the desired ideal location. The preparation and measurement cases use the PPP, PCP, and MCP instead. The same argument applies when some trailing FTECs are absent.

Definition. A set of locations in an exRec is *benign* if faults confined to those locations cannot make the exRec incorrect, for any allowed choices of errors there and elsewhere in the simulation. It is *malignant* if some choice makes it incorrect. A malignant set is *minimal* if none of its proper subsets is malignant.

Corollary. Every set of at most t locations in an exRec is benign.

⋆ Good implies correct, but bad does not imply incorrect. "Good" counts faults; "correct" describes the logical action. Counting malignant sets takes advantage of fault patterns that remain harmless even when there are more than t faults.

Theorem. If every exRec in a fault-tolerant simulation is good, or every restricted fault path is benign, the output distribution is identical to that of the ideal circuit.

Proof sketch. Start at the final measurements. Correctness replaces each measurement gadget by ideal decoding followed by ideal measurement. Move the decoders backwards

through the correct rectangles, replacing them by ideal gates. At the beginning, preparation correctness removes the remaining decoders. What remains is precisely the ideal circuit.

13.3 Bad Extended Rectangles and the $*$-Decoder

To handle a bad exRec, we would like to replace it with one faulty logical location. Two difficulties arise:

1) Overlapping exRecs can be made bad by the same faults in a shared FTEC. Counting them separately would charge those faults twice and spoil the improvement in the error exponent.

2) The logical error produced by a bad exRec can depend on the syndrome entering it. The ordinary ideal decoder discards this syndrome, so it loses information needed to describe the effective noise.

Definition. The $*$-*decoder* is a coherent version of ideal decoding that retains the syndrome register. Choosing a correctable representative E_a for each syndrome a, it acts as

$$\mathscr{D}_*(E_a\big|\overline{\psi}\big\rangle) = |\psi\rangle \otimes |a\rangle\,.$$

It extends to a unitary on the full block Hilbert space. Tracing out the syndrome register gives the ordinary ideal decoder.

The representative convention matters: a general error in syndrome class a can also include a logical operator, whose action appears in the decoded data register.

For a correct rectangle, the decoded logical action is L and is independent of the syndrome. The syndrome may change in an arbitrary way. For a bad exRec, conjugating its action by $\mathscr{D}_*$ and $\mathscr{D}_*^{-1}$ gives a noisy operation on the logical qubit and syndrome together. The retained syndromes can then be treated as part of a persistent environment.

$\star$ This is why adversarial noise appears naturally even if the physical noise was independent. The effective faults can interact through syndrome information left over from earlier bad exRecs.

13.3.1 Truncation

Procedure. Given a fixed fault path:

1) Start with the final layer of exRecs and classify each as good or bad.

2) Move backwards one layer. Remove any trailing FTEC shared with a later exRec already classified as bad.

3) Classify each resulting full or truncated exRec by the faults in the locations that remain.

4) Repeat backwards through the circuit.

The same procedure can use benign and malignant sets instead of good and bad exRecs.

⋆ Truncation is part of the proof, not a change to the physical circuit. It assigns shared FTECs consistently so that distinct effective faults require disjoint sets of physical faults.

13.4 Bounding the Logical Error Rate

Proposition. In an adversarial stochastic model, an exRec containing A locations obeys

$$\Pr(\text{bad}) \leq \sum_{\substack{S \text{ in exRec} \\ |S|=t+1}} \prod_{L \in S} p_L \,.$$

For equal error parameters $p_L = p$,

$$\Pr(\text{bad}) \leq \binom{A}{t+1} p^{t+1} \,.$$

Proof. A bad exRec contains some set of $t+1$ faulty locations. Apply the local stochastic bound to each such set and take a union bound. Fault paths with more than $t+1$ faults are already included; no additional sum over larger sets is needed.

If M_r is the number of minimal malignant sets of size r, a more careful bound is

$$\Pr(\text{malignant}) \leq \sum_{r=t+1}^{A} M_r p^r \,.$$

If only M_{t+1} is counted explicitly, one safe bound is

$$\Pr(\text{malignant}) \leq M_{t+1} p^{t+1} + \binom{A}{t+2} p^{t+2} \,.$$

The second term overcounts all possible larger minimal malignant sets, but this is acceptable for an upper bound.

13.4.1 Post-Selection

Post-selection changes probabilities. If an ancilla is used only when it passes verification, the relevant error probability is conditioned on acceptance:

$$\Pr(\text{bad} \mid \text{accept}) = \frac{\Pr(\text{bad} \cap \text{accept})}{\Pr(\text{accept})} \,.$$

Thus, a bound on the numerator alone is insufficient.

If a preparation attempt with N locations always accepts in the absence of faults, the union bound gives $\Pr(\text{accept}) \geq 1 - Np$. One can divide by this lower bound when analyzing the accepted attempt. Several post-selections require corresponding acceptance bounds, with care about conditioning and correlations.

Alternatively, include a fixed number of parallel preparation attempts in the gadget and use the first accepted ancilla. With $t+1$ attempts and at most t faults, at least one attempt

is faultless and hence accepts. Specify a fallback if all attempts reject so that the gadget is defined for every fault path. This is simple to analyze but inflates the location count; malignant-set counting can avoid charging faults in unused attempts as if they affected the data.

⋆ The assumption that faultless preparation always accepts is essential here. It does not hold for every magic state distillation protocol; for example, the five-qubit protocol accepts perfect inputs with probability only $\frac{1}{6}$.

13.5 Level Reduction

Theorem. *Level reduction.* Let C begin with preparation and end with measurement, and let $\mathsf{FT}(C)$ use a code encoding one qubit per block. Under adversarial stochastic physical noise, its output distribution equals that of a noisy realization of C with adversarial stochastic error parameters satisfying

$$p'_L \le \sum_{\substack{S \text{ minimal malignant} \\ \text{for the exRec of } L}} \prod_{M \in S} p_M \,.$$

Proof sketch. Fix a fault path and apply the backwards truncation procedure. Use $*$-decoders to replace correct exRecs by ideal locations and bad exRecs by noisy locations coupled to the syndrome environment. For any specified set of bad logical locations, truncation gives disjoint physical malignant witnesses. The probability of all these witnesses is bounded by the product of their physical error parameters. Summing over the possible witnesses gives the product of the displayed bounds, precisely the adversarial stochastic condition at the logical level.

⋆ Bounding each logical fault probability separately would not suffice. Level reduction also bounds the joint probability of faults at *any specified set* of logical locations. That is what allows the theorem to be applied again at the next level of concatenation.

The one-logical-qubit-per-block hypothesis avoids having one faulty block act on several logical qubits that belonged to separate locations in C.

13.6 Concatenation and the Threshold Theorem

Let $C_0 = C$ and $C_{\ell+1} = \mathsf{FT}(C_\ell)$. This is a simulation of a simulation, corresponding to a concatenated code. The gadget and FTEC structure repeats at each level.

For a code correcting $t \ge 1$ errors, let A bound the number of locations in any exRec and put

$$c = \binom{A}{t+1}, \qquad p_T = c^{-1/t} \,.$$

Level reduction gives

$$p_{\ell+1} \le c p_\ell^{t+1} = p_T \left(\frac{p_\ell}{p_T} \right)^{t+1} .$$

Consequently,

$$p_\ell \le p_T \left(\frac{p}{p_T} \right)^{(t+1)^\ell}.$$

For $p < p_T$, the error bound decreases doubly exponentially with the number of levels.

Proof sketch. *Threshold theorem.*

1) Let the ideal circuit contain T locations. Choose ℓ so that $Tp_\ell \le \varepsilon$. For $Tp_T > \varepsilon$, it is sufficient to choose

$$\ell \ge \log_{t+1} \left(\frac{\log(Tp_T/\varepsilon)}{\log(p_T/p)} \right),$$

rounded up to a nonnegative integer.

2) A union bound gives probability at most ε of any fault in the level-reduced circuit. If there is no such fault, the output distribution is ideal. Hence its statistical distance from the ideal output is at most ε.

3) Let B be the number of locations in the largest rectangle. Each level multiplies the size by at most B, so the physical circuit has at most TB^ℓ locations.

4) Since $\ell = O(\log\log(T/\varepsilon))$ for fixed $p < p_T$, the overhead satisfies

$$B^\ell = O\left(\left[\frac{\log(Tp_T/\varepsilon)}{\log(p_T/p)} \right]^{\log_{t+1} B} \right) = O\Big(\operatorname{polylog} \frac{T}{\varepsilon} \Big).$$

For a distance-three code, $t = 1$, and this becomes

$$p_T = \binom{A}{2}^{-1}, \qquad p_\ell \le p_T \left(\frac{p}{p_T} \right)^{2^\ell}.$$

$\star$ The overhead grows exponentially with concatenation level, but the error decreases doubly exponentially. This difference is the source of the polylogarithmic overhead.

13.6.1 Better Threshold Bounds

Theorem. A sufficient threshold condition for a fixed concatenated protocol is

$$\sum_{r=t+1}^{A} M_r p^r < p$$

for every exRec type, where M_r counts its minimal malignant sets. A threshold lower bound is obtained from the smallest positive crossing of these bounds with p.

The location-counting expression for p_T is only a lower bound on what the protocol can tolerate. A better analysis of the same gadgets, or better gadgets for the same code, may yield a larger threshold.

Definition. The *threshold of a family of fault-tolerant protocols* is the largest error rate below which that family can simulate arbitrarily long computations with arbitrarily small output error and polylogarithmic overhead. The *threshold of a code* optimizes over the relevant protocols using that code. The threshold for fault-tolerant computation optimizes over codes and protocols, for a specified error model and set of allowed operations.

⋆ An upper bound such as $p_{\ell+1} \leq c p_\ell^{t+1}$ failing to decrease above p_T does not prove that error correction fails there. It only means this particular bound no longer proves improvement.

The threshold is an asymptotic statement. A physical error rate just below threshold can still demand enormous overhead for a useful computation. Both the target logical error and the ratio p/p_T matter, as is explicit in the denominator $\log(p_T/p)$ above.

13.6.2 Threshold Surfaces and Pseudothresholds

When location types have different error rates, level reduction gives a vector recurrence

$$\mathbf{p}_{\ell+1} \leq \mathbf{f}(\mathbf{p}_\ell)\,.$$

The region whose iterates approach zero gives a sufficient region below the threshold surface. The relative sizes of the error rates generally change after a level of encoding.

Definition. A *pseudothreshold* is a crossing at which a particular logical error rate for a specified finite level of encoding equals the corresponding physical error rate, along a specified choice of physical noise parameters.

⋆ Improvement at one level is not enough to establish an asymptotic threshold. CNOT error can decrease while single-qubit error increases, and the increased single-qubit error can then make both worse at the next level. The whole vector recurrence matters.

Likewise, a relation such as $p_S = p_G/10$ imposed on physical noise need not hold between the corresponding logical error rates. Reimposing it after each level would analyze a different model.

14 Now Hold On Just a Second

Assumptions Re-Examined

14.1 Solovay-Kitaev Theorem

An ideal circuit may use gates outside the finite universal gate set implemented by our gadgets. Approximating each gate introduces another error that must be included in the accuracy budget.

Proposition. If $\|U_i - V_i\| \leq \delta_i$ for unitary gates U_i, V_i, then

$$\|U_T \cdots U_1 - V_T \cdots V_1\| \leq \sum_{i=1}^{T} \delta_i\,.$$

Proof sketch. Insert and subtract products in which one additional U_i is replaced by V_i, then apply the triangle inequality and unitary invariance of the operator norm.

Theorem. *Solovay-Kitaev.* Let S be a finite universal gate set for $\mathsf{SU}(D)$, closed under inverses, with D fixed. For any $U \in \mathsf{SU}(D)$ and $\delta > 0$, there is a product of $m = O(\text{polylog}(1/\delta))$ gates from S approximating U within operator norm δ. Such a sequence can be found classically in time $O(\text{polylog}(1/\delta))$.

Choosing $\delta = O(\varepsilon/T)$ makes the total synthesis error $O(\varepsilon)$. The additional gate count is polylogarithmic in T/ε, so combining synthesis with fault tolerance preserves the polylogarithmic overhead form.

For single-qubit Clifford+T synthesis, the approximation length can be improved to $O(\log(1/\delta))$. Fixed-size multi-qubit gates can first be decomposed into a constant number of CNOTs and single-qubit gates.

$\star$ The dimension is fixed in the Solovay-Kitaev statement. It does not say that a generic n-qubit unitary can be implemented with a number of gates polynomial in n.

14.2 Short-Range Gates

A general circuit can be converted into a nearest-neighbor circuit by routing qubits with SWAPs. For a circuit on n qubits with T locations, a crude bound is $O(n^3T)$ locations after routing, including the wait locations needed when parallel operations cannot all be routed simultaneously.

However, directly inserting SWAPs inside a fault-tolerant gadget need not preserve fault tolerance. A faulty SWAP between two qubits in the same block can introduce two errors. An ideal SWAP merely moves errors rather than multiplying them, but the faulty gate itself remains dangerous.

Procedure. *Safe exchange using an ancilla.* Route two data states through a scratch ancilla so that each physical SWAP exchanges a data state with the scratch state, never the two data states directly. On a triangle of positions i, A, j, use, in time order,

$$\text{SWAP}_{i,A}, \quad \text{SWAP}_{i,j}, \quad \text{SWAP}_{A,j} \ .$$

After the first step the scratch state is at i, and after the second it is at j. The final data states are exchanged and the scratch state returns to A.

A single faulty SWAP affects at most one data state and the scratch state. Later ideal SWAPs only move those errors, so they do not contaminate the other data state. Other geometries use extra scratch positions and additional SWAPs.

For concatenated fault tolerance, arrange each block together with the ancillas required for its gadgets in a fixed local region. A gadget between neighboring blocks then has a routing cost bounded independently of the total computer size. Concatenating this local construction preserves a threshold, though the extra locations worsen its numerical value. The initial routing of an arbitrary nonlocal algorithm is a separate cost from the fault-tolerance overhead for the resulting local circuit.

⋆ Locality must be built into the gadgets at each level. Routing across the whole growing computer inside every gadget would not give a constant-size replacement at each level.

14.3 Slow Measurement or No Intermediate Measurement

If measurements or classical decoding are slow, data may accumulate storage errors while waiting for the result. A fixed delay can be absorbed into the gadget, but this can unnecessarily reduce the threshold.

14.3.1 Coherent Classical Processing

Intermediate measurements can be deferred by retaining their outcomes in quantum registers and doing the subsequent classical computation reversibly. However, classical processing was assumed perfect, while these new quantum gates are noisy. The replacement computation must therefore be fault-tolerant too.

The apparent circularity is resolved because these registers contain *classical* information. Only bit flip errors matter; phase errors on a syndrome register that is eventually discarded do not change its classical value. Repetition codes and classical fault-tolerant circuits suffice.

Procedure.

1) Keep the qubits that would have been measured as syndrome registers.
2) Compute the decoded syndrome or logical outcome repeatedly into separate ancilla registers, obtaining a repetition encoding of the result.
3) Maintain these registers using classical fault tolerance implemented coherently.
4) To condition a transversal quantum operation on the result, use a different copy of the control bit for each physical data qubit.

⋆ Simply copying one possibly wrong control bit many times does not make it reliable. Nor should one noisy control qubit determine a correction on every qubit of the block. Redundant computation and distributed controls prevent a single control fault from causing many data errors.

14.3.2 Delayed Classical Outcomes

For Clifford processing, a syndrome result is often needed only to update the Pauli frame. The quantum circuit can continue while measurement and decoding finish, and the classical frame can be updated afterwards. A non-Clifford gadget may require the result before choosing a correction or measurement basis, so those dependencies still need to be respected.

Ancilla preparation and verification can also be performed in advance, with storage and error correction while waiting. At sufficiently high concatenation levels, logical gadgets are long compared with a fixed physical measurement delay. Thus, slow measurement need not remove the threshold.

14.4 Fresh Ancilla Qubits

Error correction transfers entropy from the data into syndrome ancillas. To repeat the process indefinitely, we need somewhere to put that entropy. Unitary gates alone cannot remove it from the complete system.

Theorem. Consider n qubits with arbitrary unitary control, followed at each time step by independent depolarizing noise with a fixed $0 < p \leq \frac{3}{4}$. Without fresh ancillas or another entropy-removal mechanism, the state approaches the maximally mixed state exponentially in time. In particular, useful arbitrary computation cannot persist much longer than $O(\log n)$ time steps in this model.

Proof sketch. Let $S(\rho)$ be the von Neumann entropy in bits. Since

$$\mathcal{D}_p(\rho) = \Big(1 - \frac{4p}{3}\Big)\rho + \frac{4p}{3}\frac{I}{2},$$

the entropy deficit obeys

$$n - S(\rho_{j+1}) \leq \Big(1 - \frac{4p}{3}\Big)[n - S(\rho_j)]\,.$$

Unitary control leaves entropy unchanged. Iterating gives

$$n - S(\rho_j) \leq n\Big(1 - \frac{4p}{3}\Big)^j.$$

Since this deficit equals the relative entropy from ρ_j to $I/2^n$, Pinsker's inequality implies

$$\Big\|\rho_j - \frac{I}{2^n}\Big\|_1 \leq \sqrt{2\ln 2\, n\Big(1 - \frac{4p}{3}\Big)^j}\,.$$

After a time logarithmic in n, the state is close to maximally mixed irrespective of the unitary computation.

$\star$ "Fresh" does not necessarily mean a newly manufactured qubit. A reset qubit is fresh in the relevant sense: it provides low-entropy degrees of freedom into which error correction can transfer entropy.

The conclusion depends on the noise model:

1) *Amplitude damping:* Used ancillas can relax towards $|0\rangle$, allowing them to be reused. The noise itself supplies a cooling mechanism.

2) *Non-unital noise:* A non-maximally mixed reset state can be useful even if it is not nearly pure. Algorithmic cooling can concentrate entropy in some qubits and extract colder ancillas from the others, with additional overhead and sufficiently accurate control.

3) *Pure dephasing:* Ancillas initially prepared in computational basis states remain usable until needed. Used ancillas cannot be reset by dephasing alone, so a sufficiently large supply must be prepared at the start. Arbitrarily long computation can still be possible below threshold, but with polynomial rather than polylogarithmic overhead in this setting.

Definition. *Algorithmic cooling* uses unitary operations to redistribute entropy, producing some lower-entropy qubits at the expense of higher entropy in the remaining qubits. Unitary cooling alone does not reduce total entropy; repeated reset of the hotter qubits supplies the entropy sink.

14.5 Parallelism and Timing

With nonzero storage error rate, a large computer needs parallel error correction. If at most c qubits can be acted on per time step, touching all n qubits takes at least $\lceil n/c \rceil$ steps. A qubit waiting that long avoids independent storage faults with probability only

$$(1 - p_S)^{\lceil n/c \rceil},$$

which tends to zero as n grows with fixed c.

Maximal parallelism is not necessary. Acting on a constant fraction of the qubits per step adds only a constant number of waits. For instance, operating on one-third of the qubits at once can be modeled by adding up to two wait locations per gate, changing the effective rates roughly as

$$p_S \longmapsto 3p_S, \qquad p_G \longmapsto p_G + 2p_S\,,$$

to first order in small error rates.

If different gates take different times, one simple schedule pads the shorter gates with waits to match the longest. More careful schedules may save time, but must respect dependencies and prevent some blocks from waiting arbitrarily long for others. Storage faults belong in the location count just as gate faults do.

14.6 Leakage Errors

Definition. A *leakage error* takes a physical qubit outside its intended two-dimensional computational subspace. Loss of a particle is one example; population of unwanted internal levels is another.

A leaked qubit can cause additional errors whenever it interacts with other qubits, so merely measuring the ordinary syndrome may not be enough. The qubit must be returned to the computational subspace or replaced.

If leakage is detected and its location known, replacing the affected qubit converts the problem into an erasure. If the location is not known, replacement still helps, but the residual error must generally be treated as an unknown-location error.

⋆ Knill EC and gate teleportation automatically transfer the state to fresh physical qubits. Leakage in an incoming qubit can affect the Bell measurement, but it does not persist as leakage of that same particle in the output block. Fresh output qubits can still have new faults of their own.

Frequent teleportation can provide this replacement even in a protocol that otherwise uses a different FTEC. Leakage monitoring and teleportation are compatible: monitoring can

supply erasure information, while teleportation prevents leaked particles from remaining in the data indefinitely.

14.7 Biased Errors

If phase errors are much more common than bit flips, a code that spends more redundancy on phase protection may be more efficient. CSS codes permit different protection against the two error types. However, a computation changes errors as well as data.

For example, $HZH = X$ turns a common phase error into a bit flip. A gate set that repeatedly changes the noise bias can defeat a code chosen specifically for dephasing.

Definition. A gate implementation is *bias-preserving* for a specified error model if the dominant errors retain their favored form under the implementation, including errors occurring during the gate.

⋆ It is not enough to check conjugation by the completed gate. An error can occur halfway through its physical implementation. CNOT preserves the set of Z-type Paulis under its final conjugation action, but an implementation on strictly two-level systems need not preserve dephasing bias throughout the gate.

Diagonal Hamiltonians commute with Z errors, making diagonal gates such as controlled-Z natural ingredients for dephasing-biased protocols. Preparation of $|+\rangle$ and measurement in the X basis also fit this setting. Logical gates can then be assembled using measurements, teleportation, and suitable resource states.

For the phase repetition code with codewords $|+\rangle^{\otimes n}$ and $|-\rangle^{\otimes n}$, one may choose

$$\overline{X} = Z^{\otimes n}, \qquad \overline{Z} = X_1 \, .$$

Transversal X measurement followed by majority decoding measures logical Z. Logical X can be measured using a $|+\rangle$ ancilla coupled by controlled-Z to all data qubits, followed by X measurement of the ancilla and sufficient repetition. Under the pure phase-error model, phase errors on the ancilla do not spread through controlled-Z into multiple data errors.

⋆ This single-ancilla measurement is only justified by the restricted noise model. Rare bit flips can propagate badly, so a realistic biased-noise protocol must also account for them, often by concatenation with a code protecting against general errors.

With both phase and non-phase faults present, the threshold is a region in the space of the two error rates. The additional locations used to suppress phase faults can amplify the effect of the rare error types, so exploiting bias only helps when the bias is sufficiently strong and preserved by the gadgets.

14.8 Error Independence and Non-Markovian Errors

14.8.1 Spatially Correlated Errors

The number of possible correlated partners matters as much as the strength of an individual pair interaction. If every pair of qubits fails with the same small probability, the total error

rate involving any one qubit grows with the size of the computer.

Suppose the probability of a correlated error between qubits separated by distance r scales as $pr^{-\alpha}$ in D spatial dimensions. The total contribution per qubit is controlled by

$$p \int_1^{n^{1/D}} r^{D-1-\alpha}\, dr\,.$$

This is bounded independently of n when $\alpha > D$, grows logarithmically when $\alpha = D$, and grows as a power when $\alpha < D$. Under appropriate independence assumptions between the pair-error events, the convergent case can be bounded by an adversarial stochastic model, though with a worse effective error parameter.

$\star$ A small error rate measured on a small device is not by itself the required scalability condition. The noise model must remain sufficiently weak per qubit as the number of qubits grows.

14.8.2 Coherent Fault Paths

A small coherent overrotation is not a rare large fault. For example,

$$e^{-\iota\theta Z} = \cos\theta\, I - \iota \sin\theta\, Z$$

contains identity and error amplitudes in superposition. In a circuit, the expansion is a sum over fault paths rather than a probabilistic mixture. Different paths can interfere, so the proof must bound norms of sums of faulty contributions instead of simply adding fault probabilities.

14.8.3 Hamiltonian Noise Model

Definition. A *non-Markovian error model* is described by

$$H = H_S + H_B + H_{SB}\,,$$

where H_S acts on the computer and implements the ideal circuit, H_B acts on the bath, and H_{SB} couples the system to the bath and includes control errors. A *k-local* model has

$$H_{SB} = \sum_{\substack{T \text{ a set of system qubits} \\ |T| \le k}} H_T\,,$$

where H_T acts on the system qubits in T and arbitrary bath degrees of freedom. The noise strength is the maximum operator norm $\max_T \|H_T\|_\infty$, uniformly over time. The Hamiltonians may be time-dependent.

Use units in which a computational time step is 1 and $\hbar = 1$. With physical gate duration τ, the corresponding dimensionless coupling scale is $\tau\|H_T\|_\infty/\hbar$.

Theorem. There is a positive threshold for sufficiently weak 1-local non-Markovian noise. Below it, an ideal circuit with T locations can be simulated to output statistical distance at most ε with overhead polylogarithmic in T/ε.

Lemma. For 1-local noise of strength at most η, and ideal gates acting on at most m qubits per location, the joint system-bath evolution has a fault-path expansion

$$U = \sum_{\Omega} W_{\Omega} \,,$$

where locations outside Ω act ideally. For a specified set S of faulty locations, let

$$E_S = \sum_{\Omega \supseteq S} W_{\Omega} \,.$$

Then

$$\|E_S\|_{\infty} \leq (m\eta)^{|S|} \,,$$

and the same bound holds for a fixed exact fault-path contribution W_S.

Proof sketch. Divide each computational time step into intervals of length δ. The noise evolution in one interval is

$$e^{-\iota\delta H_{SB}} = I - \iota\delta \sum_{a} H_a + O(\delta^2) \,.$$

A faulty location contains at least one non-identity noise insertion. Summing over its possible first insertion times gives a factor bounded by $m\eta$. The remaining evolution can be grouped into unitary factors of norm 1. Repeating this for each specified faulty location and taking $\delta \to 0$ gives the bound.

The threshold proof again groups paths by bad exRecs and performs level reduction, now using operator-norm bounds and the triangle inequality. Retaining the bath and syndrome registers allows correlations and entanglement between different times. The bath need not forget between gates.

$\star$ The bound must apply to the sum over paths containing S, not just each exact path separately. There are exponentially many paths, and bounding all of them independently can lose control of the total error.

The 1-local theorem does not directly include arbitrary simultaneous errors on both qubits of a two-qubit gate. Extensions allow multi-qubit noise terms tied to gate locations, or sufficiently decaying correlated interactions, but their hypotheses must be included explicitly.

14.8.4 Error Amplitudes Versus Error Probabilities

For a stochastic error of probability p, a purified system-bath description can have an error amplitude of order $\sqrt{p}$. Thus, a threshold expressed as a Hamiltonian norm or amplitude bound cannot be numerically compared with a stochastic probability threshold without converting the parameters.

Coherent errors can add constructively: T identical small overrotations can accumulate an angle proportional to $T\theta$. Random phases instead tend to produce a square-root accumulation. Gates change the directions of errors, and syndrome measurements decohere fault paths

with different syndrome records. Consequently, the worst coherent bound can be much more pessimistic than a particular physical noise process.

⋆ Syndrome measurement does not eliminate all coherence between errors. Fault paths producing the same full syndrome record can still interfere. Treating every coherent noise process as a stochastic Pauli channel without justification would discard this effect.

Pauli twirling can average a channel into a Pauli channel, but physical twirling gates have their own faults. A proof that assumes perfect randomization must account for those faults before being applied to an implementation.

14.8.5 Limits of the Assumptions

A small operator-norm bound is a sufficient condition, but it can fail for noise that is weak on the bath states actually occupied. A rare bath excitation may interact strongly with the system, or an infinite-dimensional bath may give an unbounded interaction Hamiltonian even at low temperature. Such cases require additional information about the bath state and dynamics.

⋆ Failure of the hypotheses of a threshold theorem is not a proof that fault tolerance is impossible. Conversely, a theorem for a specified noise model does not establish that a physical device obeys that model. The assumptions about noise strength, correlations, entropy removal, and available control are part of the result.