1 Today's Outline

- Recap of Shor code
- Stabilizer code
- CSS codes
- Analyzing distance of quantum codes

2 Recap of Shor code

2.1 Definition

We covered the Shor code in the previous lecture. This code encodes a single qubit and simultaneously corrects both bit flip and phase flip errors. The code is defined by its encoding of the two states $|0\rangle|0\rangle^8$ and $|1\rangle|0\rangle^8$ where the last 8 bits are ancilla bits.

$$|0\rangle|0\rangle^8 \to |0\rangle_{shor} = \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 3}$$
$$|1\rangle|0\rangle^8 \to |1\rangle_{shor} = \left(\frac{|000\rangle - |111\rangle}{\sqrt{2}}\right)^{\otimes 3}$$

We can list 8 *stabilizers* for this code. A stabilizer for this code is an operator which acts as the identity on both $|0\rangle_{shor}$ and $|1\rangle_{shor}$ (and therefore on any encoded qubit).

$$\begin{aligned}
P_1 &= Z_1Z_2, \quad P_2 = Z_2Z_3,\\
P_3 &= Z_4Z_5, \quad P_4 = Z_5Z_6,\\
P_5 &= Z_7Z_8, \quad P_6 = Z_7Z_8,\\
P_7 &= X_1X_2X_3X_4X_5X_6,\\
P_8 &= X_4X_5X_6X_7X_8X_9.
\end{aligned}$$

In fact, the Shor code can be defined in terms of these stabilizers $P_1 \dots P_8$. In other words, the Shor code is $\{|\psi\rangle : P_i|\psi\rangle = |\psi\rangle\}$.

2.2 Correcting One Error

Next, we describe how to correct an error with this code. Recall that it should correct a single bit flip or a single phase flip. Let $|\psi\rangle$ be the initial state, consisting of 9 qubits. Let E be the error operator, which acts on $|\psi\rangle$, so that the corrupted state is $E|\psi\rangle$. Then $E = I$ if there is no error, and otherwise it is X_j or Z_k for some indices j, k. For each i, we have a separate ancilla bit starting as $|0\rangle_i$ and take the map that sends

$$E|\psi\rangle|0\rangle_i \to \frac{I+P_i}{2}E|\psi\rangle|0\rangle_i + \frac{I-P_i}{2}E|\psi\rangle|1\rangle_i \tag{25.1}$$

When $E = I$ the state doesn't change, so P_i is a stabilizer. As such, this operation preserves the original state and the ancilla bit stays $|0\rangle_i$. If, on the other hand, $P_i(E|\psi\rangle) = -E|\psi\rangle$, then the expression would preserve the state bits while changing the ancilla bit to $|1\rangle_i$. Thus, if we can show

- When $E = X_j, Z_k$, a single bit flip or phase flip causes $P_i(E|\psi\rangle) \in \{-E|\psi\rangle, E|\psi\rangle\}$ for all i (resulting in the corresponding ancilla bit being $|0\rangle_i$ or $|1\rangle_i$).
- For at least some i, we have $P_i(E|\psi\rangle) = -E|\psi\rangle$.

then we can detect an error. The first statement holds because X_j and X_k always commute or anti-commute. As such, applying E and then P_i or P_i and then E can only differ in a factor of -1 since both expressions are comprised purely of X_j and Z_k operators. The second statement holds because the elements of the Shor code are the only simultaneous stabilizers of all the P_i.

The value of the ancilla bits after performing the operation for all the values of i is known as the *syndrome*. If we could show that each possible phase flip or bit flip resulted in a different syndrome, we would be able to correct the error as well. So let's write out the syndrome for each P_i for each of $E \in \{X_1 \dots X_8, Z_1 \dots Z_8\}$. This is 0 if P_i commutes with E and 1 if it anti-commutes.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
X_1	1	0	0	0	0	0	0	0
X_2	1	1	0	0	0	0	0	0
X_3	0	1	0	0	0	0	0	0
X_4	0	0	1	0	0	0	0	0
X_5	0	0	1	1	0	0	0	0
X_6	0	0	0	1	0	0	0	0
X_7	0	0	0	0	1	0	0	0
X_8	0	0	0	0	1	1	0	0
X_9	0	0	0	0	0	1	0	0

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
Z_1	0	0	0	0	0	0	1	0
Z_2	0	0	0	0	0	0	1	0
Z_3	0	0	0	0	0	0	1	0
Z_4	0	0	0	0	0	0	1	1
Z_5	0	0	0	0	0	0	1	1
Z_6	0	0	0	0	0	0	1	1
Z_7	0	0	0	0	0	0	0	1
Z_8	0	0	0	0	0	0	0	1
Z_9	0	0	0	0	0	0	0	1

So if the error is any of the X_j, then the syndrome uniquely determines what error happened. Then, the operator for that error can be re-applied to get back the uncorrupted qubits.

The only cases where the syndrome does not uniquely determine the error is differentiating between Z_1, Z_2, Z_3 or Z_4, Z_5, Z_6 or Z_7, Z_8, Z_9. However, it turns out that applying the fix Z_1 works *regardless* of whether the error was Z_1, Z_2, Z_3 (and similarly for Z_4, Z_5, Z_6 or Z_7, Z_8, Z_9). This concept is known as *degeneracy*.

3 Stabilizer Codes

The Shor code is an example of a stabilizer code. The phenomena described above can be generalized, by defining a code based on some different set of stabilizers.

Definition 25.1 (Pauli Group)**.** The Pauli Group $\mathcal{P}_n$ is the group generated by the operators $\{X_j\}_{j\in[n]}$,$\{Y_j\}_{j\in[n]}$ and $\{Z_j\}_{j\in[n]}$

Because $Y_j = iX_jZ_j$, only the X and Z operators are relevant, up to constants.

Definition 25.2 (Stabilizer Code)**.** Let H be an abelian subgroup of $\mathcal{P}_n$ and let $P_1 \dots P_m$ generate G. Then the code is

$$\{|\psi\rangle \in (\mathbb{C}^2)^n \ |P_i|\psi\rangle \text{ for } i = 1 \dots m\}$$

Then, similar to above, one can apply the operator E to $|\psi\rangle$ where E is a possible error pattern, and then check the syndrome upon applying the transformation described by Equation (**??**). As long as one can show all the error patterns give different syndrome, or are equivalent up to degeneracy, the code would correct for the given errors.

3.1 Parameters

In classical codes, an $[n, k, d]$ code has the following parameters: k is the length of the original message, n is the length of the encoded message, and d is the distance of the code. The parameters n and k still make sense in the quantum context, while distance needs to be revisited.

In classical codes, the distance between two codewords can be thought of as the number of bit flips needed to get from one to the other. In quantum coding, we similarly define the distance to be the minimum number of X, Z, or Y transformations to get from one codeword to another. In other words, we want the minimum weight E such that $E|\psi\rangle \in \text{Stab}_G \setminus \{|\psi\rangle\}$. The only way to get such an E is for it to be in $N(H)$ where $N(H)$ is the normalizer. As such, the distance of a stabilizer code is

$$d = \min_{E\in N(H)\setminus H}\{wt(E)\}.$$

One can try to compute the distance of the Shor code and show it's 3 (like a classical code correcting one bit flip would be). We need to find $N(P)$ where P is the subgroup of $\mathcal{P}_9$ formed by the 8 generators of the Shor code described earlier. Our strategy will be to see what subset of $X_1 \dots X_9, Y_1 \dots Y_9, Z_1 \dots Z_9$ that E contains. The ordering doesn't matter, since all X, Y, Z commute or anti-commute (so changing the ordering doesn't affect whether it commutes with P).

Though we won't demonstrate fully how this argument is done, one can show, for example, that for E to commute with Z_1Z_2, that E must have all or none of X_1, X_2, X_3 if E has only X's. If it has no X's, then every block of Z's (Z_1, Z_2, Z_3 and Z_4, Z_5, Z_6 and Z_7, Z_8, Z_9) must have the same parity of things included. Using more arguments like this, one can narrow down what $N(P)$ looks like, and show all elements are at least weight 3.

4 Calderbank-Shor-Steane (CSS) Codes

CSS codes are a particular type of stabilizer code. Specifically, they are stabilizer codes where each P_i has only X_j's or Z_j's in it. Note that the Shor code is an example of a CSS code.

The set M_X denotes the subset of P_i's with X_j's and M_Z denotes the subset with Z_j's. The matrix H_X is the parity check view of the M_X Z-stabilizer, and H_Z is defined similarly. In the case of the Shor code, H_X looks like

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

where the rows correspond to $M_X = \{P_7, P_8\}$ and the columns to $Z_1 \dots Z_9$, and the entries are whether they commute.

This code is a $[n, n-|M_X|-|M_Z|, ?]$ code. Let's try to pin down the distance. We're looking for the normalizer of H, and the matrices H_X and H_Z tell us exactly what each stabilizer commutes with. We define $C_X = \ker(H_X)$ and $C_Z = \ker(H_Z)$. Then, it holds that $d \geq \min\{d(C_X), d(C_Z)\}$. Now, C_X and C_Z are classical codes – so we've reduced our quantum coding problem to two classical coding problems! We simply need to find the distance of C_X and C_Z to bound the distance of the quantum code. In fact, CSS codes are often referenced in terms of these two classical codes C_X and C_Z as $\mathrm{CSS}(C_X, C_Z)$.