Stabilizer Groups for Quantum Error Correction

Rushil Shah
Department of Physics, 77 Massachusetts Ave., Cambridge, MA 02139-4307
(Dated: May 22, 2026)

We introduce the notion of a stabilizer code, provide an example of using it for quantum error correction, and understand why it is a promising area of research for quantum computation.

I. INTRODUCTION

In the past few decades, the field of quantum computing has exploded. Algorithms such as Shor's Algorithm for factorization have grown popular and research groups and companies are investing heavily in this future technology. However, there are many challenges preventing quantum computing from replacing classical computing.

One of the main challenges is the relatively low fidelity of physical quantum gates and qubits. Currently, some of the highest fidelity quantum gates ever manufactured [6] still pale in comparison to the fidelity of transistors, the main component of classical gates. To overcome this major issue, substantial effort has been invested into quantum error correction.

Quantum error correction (QEC) focuses on identifying errors in qubit states and correcting them to overcome the high fault rates of real quantum gates. In this paper, we will begin by reviewing quantum computing fundamentals and introduce the simple repetition three qubit code. Then, we will introduce a promising quantum error correcting code formulation called the stabilizer code, providing both mathematical and physical intuition on how it corrects errors. Using this formulation, we will apply it to the simple repetition code and then to the five qubit perfect code. Finally, we will cover some challenges that stabilizer codes face and current areas of research advancing the applications of these special codes.

A. Quantum Computing Overview

We can represent a qubit as a superposition of the classical 0 or 1 states,

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle = \begin{pmatrix} a \\ b \end{pmatrix}.$$

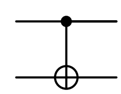
where $a^2 + b^2 = 1$. This superposition is the key to what provides quantum circuits a computational advantage to classical circuits. We can also visualize how quantum gates change the state of qubits. Some of the most popular single qubit states are,

$$-\boxed{Z}- = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad -\boxed{X}- = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$
$$-\boxed{Y}- = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad -\boxed{H}- = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

We can think of gates as operators acting on a quantum state (qubit), transforming the qubit into a desired state. Notice that the X, Y, and Z gates are identical to the Pauli spin operators. The H gate, Hadamard gate, in particular is interesting because it creates a superposition from a $|0\rangle$ or $|1\rangle$ state,

$$\hat{H}\,|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

There also exist two and more qubit gates which take in $N \geq 2$ qubits and output N qubits. The most popular two qubit gate is the CNOT gate,



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

Intuitively, the CNOT gates takes a control qubit, the top qubit, such that if it is 0, it doesn't do anything to the bottom qubit. However, if it is 1, then it performs an X-gate on the second qubit. These control operators are common in quantum circuits, allowing for entanglement and more complex interactions as we will see in the quantum decoding circuit.

B. A Simple Error Correcting Code

Now that we have covered the fundamentals of quantum computing, let's introduce the simplest quantum code and visualize the effect of an error. Suppose we have a qubit $|0\rangle$. If we wanted to detect any errors, we could try duplicating so that we have the state $|0\rangle \otimes |0\rangle \otimes |0\rangle$ where we have three qubits tensored. If there was an error on one of these qubits, turning a 0 into a 1, then we would have 2 0-qubits and 1 1-qubit, so by majority, we can say that the qubit represents a 0-qubit. We can state this code more formally as follows,

$$|0\rangle_L = |0\rangle \otimes |0\rangle \otimes |0\rangle = |000\rangle\,, \tag{1}$$
$$|1\rangle_L = |1\rangle \otimes |1\rangle \otimes |1\rangle = |111\rangle\,, \tag{2}$$

so the code space is spanned by $|000\rangle$ and $|111\rangle$. This code is duplicating the 0 and 1 states of the qubit creating a logical 0 qubit, $|0\rangle_L$, and logical 1 qubit, $|1\rangle_L$. Therefore, we can represent the repeated 0 qubits with a logical 0 qubit and same for 1.

To better understand the effect of an error on this code, let's consider an X error on the first qubit of a quantum state. Recall that an X operator makes a 0 state a 1 state and vice versa,

$$(X \otimes I \otimes I)\,|\psi\rangle = (X \otimes I \otimes I)(a\,|000\rangle + b\,|111\rangle) = (a\,|100\rangle + b\,|011\rangle) \neq a\,|0\rangle_L + b\,|1\rangle_L .$$

However, if we use the majority rule, we see that the first state has majority 0s, so we can correct it to the $|0\rangle_L$ state and similarly, the second state, $|011\rangle$, has majority 1s, so we can correct it to the $|1\rangle_L$ state. This procedure, although not mathematically rigorous, allows us to correct for an X-error on the state.

Unfortunately, there are some setbacks to this code. Suppose that there was an X-error on two qubits, then the state with the error will be,

$$X_1 X_2\,|\psi\rangle = a\,|110\rangle + b\,|001\rangle ,$$

so the corrected state will become

$$|\psi\rangle = a\,|111\rangle + b\,|000\rangle .$$

So, this code only allows us to correct 1 X-error.

Furthermore, consider the action of a Z-error,

$$(Z \otimes I \otimes I)\,|\psi\rangle = a\,|000\rangle - b\,|111\rangle \neq |\psi\rangle ,$$

since the Z operator creates a phase for the 1 state. However, due to this phase, the resulting state isn't the same, so the code can't correct for these "phase flip" errors.

II. STABILIZER GROUP DEFINITION

To resolve these issues with the simple repetition code and provide a more rigorous definition of correctable errors, we will introduce the stabilizer group and its corresponding code.

$$\text{Stabilizer Def}: S\,|\psi\rangle = |\psi\rangle\,, \forall\,|\psi\rangle \in C. \tag{3}$$

As in (3), we can think of a stabilizer as any operator which doesn't change any state in the code space [2].

For those unfamiliar with group theory, it suffices to think of the stabilizer group as the minimum size set of operators such that multiplying any combination of stabilizers produces all possible stabilizers of the code.

Some important properties about the stabilizer group, the group generating the set of all stabilizing operators for a given code,

- Two stabilizer operators must commute.
- S is a subgroup of $\mathcal{G}$, the group of all Pauli operators,

  $$\mathcal{G} = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}^{\otimes n}.$$

  In other words, S only consists of operators that exist as some linear combination of Pauli operators.

Let's take a second and think about why a stabilizer code would be helpful. Suppose we have some code space, C, stabilized by a group, S, with a Pauli error E. In order for the error to affect the code, it can't be part of the stabilizer group, so there must exist some $M \in S$ such that $[M, E] \neq 0$. In fact, since both M and E are in $\mathcal{G}$, properties of Pauli matrices imply that M and E must either commute or anticommute. Therefore, if we know which M anticommute with our error, E, we can determine what our pauli error was and apply that same error to the state to correct the error because $E^2 = 1$. This motivates us to write an error syndrome for stabilizer codes [2], $f_M : \mathcal{G} \to \mathrm{Z}_2$,

$$f_M(E) = \begin{cases} 0 & \text{if } [M, E] = 0, \\ 1 & \text{if } \{M, E\} = 0. \end{cases} \tag{4}$$

In other words, we define the error syndrome of a stabilizer code as the bit string, $f(E) = 0$ if E commutes with M or 1 if E anticommutes with M for all M in the generator set of S. Therefore, by measuring each M on the state, we can generate the error syndrome to identify and correct the error on the state.

Suppose an error, E, commutes with the stabilizer, M, then measuring the stabilizer on the state with the error, $|\phi\rangle = E\,|\psi\rangle$ would yield,

$$M\,|\phi\rangle = ME\,|\psi\rangle = EM\,|\psi\rangle = E\,|\psi\rangle = |\phi\rangle ,$$

or an eigenvalue of 1. However, suppose the error anticommutes with the stabilizer,

$$M\,|\phi\rangle = ME\,|\psi\rangle = -EM\,|\psi\rangle = -E\,|\psi\rangle = -\,|\phi\rangle ,$$

or an eigenvalue of -1. Therefore, if we measure the stabilizer and get 1, then the error syndrome will have a 0 for that stabilizer, and if we receive a -1, the error syndrome will get a 1.

A. Quantum Code Notation

Before we provide an example of syndrome decoding, it is worth to introduce some notation to simplify describing quantum codes,

$$\text{Quantum Code: } [[n, k, d]].$$

In this representation, we state the number of physical qubits each code state contains as n, the number of logical qubits encoded in each code state as k, and the distance of the code as d. The distance is an equivalent way to say that the number of single qubit errors the code corrects is $\lfloor (d-1)/2 \rfloor$ [9].

This new representation of quantum codes provides a new perspective on the size of the stabilizer group. Intuitively, the more generators there are, the more restrictive the code is since there are less states that can be stabilized. So, if we have n physical qubits and we want to encode them into a smaller space, we can add independent stabilizers to restrict physical space into a code space of size k. Mathematically,

$$r = n - k, \tag{5}$$

where r is the number of generators of the stabilizer [9].

Recalling our simple repetition code from before, (1) and (2), we took three physical qubits and compressed them into one logical qubit. Therefore, $n = 3$ and $k = 1$. The distance is more tricky to understand; however, because the code couldn't correct against all 1-qubit errors, it failed to correct Z errors, we can say that the distance of the code is 1. So, we can denote the simple repetition code as a $[[3, 1, 1]]$ code.

B. Simple Repetition Stabilizer Code

Let's use the formalism we have built to understand the simple repetition code as a stabilizer code. We can begin by using (5) to determine the number of independent generators of our stabilizer group,

$$r = n - k = 3 - 1 = 2.$$

Therefore, we need to find 2 stabilizing generators of our code space. Looking at the code, we notice that applying a Z operator to any qubit only multiplies its phase by -1 if it is a $|1\rangle$ state and doesn't do anything if its a $|0\rangle$ state. Therefore, applying any amount of Z operators in any configuration will stabilize the $|000\rangle$ state. However, for the $|111\rangle$ state, we would need exactly two Z operators to cancel the phase so that it stabilizes the state. So, we claim that $Z \otimes Z \otimes I = Z_1Z_2$ and $I \otimes Z \otimes Z = Z_2Z_3$ are the stabilizer generators. The operator $Z \otimes I \otimes Z = Z_1Z_3$ is also a stabilizer, but it can be generated from the other generators,

$$Z_1Z_2 \cdot Z_2Z_3 = Z \otimes (Z \cdot Z) \otimes Z = Z \otimes I \otimes Z = Z_1Z_3.$$

Thus, the stabilizer group for the simple repetition group is generated by,

$$S = \{Z_1Z_2, Z_2Z_3\}.$$

Intuitively, this says that any operator which doesn't change a code state can be generated as a linear combination of Z_1Z_2 and Z_2Z_3. Additionally, this matches up with the expected number of independent generators, $n - k = 2$, providing extra confirmation the stabilizer group is correct.

Now, we can see why the single qubit X operator can be corrected for but the single qubit Z operator can't. Recall the stabilizer error syndrome from (4). Using this error syndrome, we know that an error can be detected and corrected for if it anticommutes with at least one of the stabilizer generators.

Suppose our error is $E = X_1 = X \otimes I \otimes I$. Calculating the error syndrome,

$$\{Z_1Z_2, X_1\} = 0, [Z_2Z_3, X_1] = 0 \implies f(E) = \{1, 0\}.$$

This means that if we measured the stabilizer generators on the state acted by X, we would get -1 for Z_1Z_2 and 1 for Z_2Z_3. Let's show this,

$$\begin{aligned} |\psi\rangle &= X_1(a\,|000\rangle + b\,|111\rangle) = a\,|100\rangle + b\,|011\rangle\,, \\ Z_1Z_2\,|\psi\rangle &= -a\,|100\rangle - b\,|011\rangle = -\,|\psi\rangle\,, \\ Z_2Z_3\,|\psi\rangle &= a\,|100\rangle + b\,|011\rangle = |\psi\rangle\,. \end{aligned}$$

This aligns with the expected result! Since the stabilizer measurements match up with the error syndrome of the X_1 operator, if we didn't know what error the state received, we could determine that the error was the single qubit error X_1. So, if we wanted to correct the error, we could then apply an X_1 gate to the state.

Now, let's take a look at where the simple repetition code falls short. Consider the error syndrome of the $Z_1 = Z \otimes Z \otimes I$ error,

$$[Z_1Z_2, Z_1] = 0, [Z_2Z_3, Z_1] = 0 \implies f(E) = \{0, 0\}.$$

For this error, there is no observable syndrome. Therefore, if we were to measure the stabilizer generators on the state, we would get an eigenvalue of 1 for both measurements, making us think that the state has no errors on it! However, we know this isn't true because a Z error on the state would result in,

$$Z_1(a\,|000\rangle + b\,|111\rangle) = a\,|000\rangle - b\,|111\rangle \neq |\psi\rangle\,,$$

a completely new state. So, the simple repetition code fails in detecting and observing Pauli Z errors.

C. Stabilizer Code Decoding Circuit

Before moving onto a more complex code, let's visualize how these error detections can be implemented in a quantum circuit.

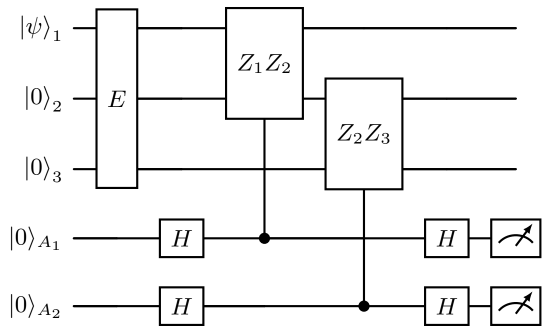


FIG. 1: Decoding Circuit for Simple Repetition Code

The circuit in Figure 1 may appear ambiguous, so let's see how this circuit works to detect error. Note that we interpret a quantum circuit left to right with each box representing some sort of operator on the qubits passed into it. The bottom two qubits are known as "ancilla qubits", representing additional qubits on the circuit typically used to store a state.

Let us consider the X_1 error from before. This means that the first qubit, top most qubit, in the circuit has an error, causing the $|0\rangle$ and $|1\rangle$ states of the input state to be swapped. Let's follow the state of the circuit over the operators. Suppose that the state after the E gate, the encoder operator combined with the X_1 error, is,

$$|\Psi\rangle = X|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle \otimes |0\rangle_{A_1} \otimes |0\rangle_{A_2}.$$

Consider the first stabilizer generator first, Z_1Z_2. Since the third qubit and the second ancilla qubit, $|0\rangle_{A_2}$, aren't affected by the stabilizer, we will ignore them when writing out the state. First, let's see the effect of the Hadamard on the state,

$$|\Psi\rangle = X|\psi\rangle \otimes |\psi\rangle \otimes H|0\rangle = X|\psi\rangle \otimes |\psi\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

The Hadamard, refer to section I A for an overview of the Hadamard, creates a superposition of the ancilla bit. To see why this is relevant, let's see the effect of the controlled Z_1Z_2 gate on the state. Note that the dot on the ancilla qubit implies it is the control bit. In other words, if the ancilla qubit is a 0, then nothing happens on the first two qubits. However, if the ancilla qubit is a 1, then Z_1Z_2 is applied on the first two qubits. If we apply this operator on the state, it becomes,

$$|\Psi\rangle = \frac{1}{\sqrt{2}}X|\psi\rangle \otimes |\psi\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}}ZX|\psi\rangle \otimes Z|\psi\rangle \otimes |1\rangle. \tag{6}$$

Finally, we can apply the Hadamard on the ancilla qubit again,

$$|\Psi\rangle = \frac{1}{2}X|\psi\rangle \otimes |\psi\rangle \otimes (|0\rangle + |1\rangle) + \frac{1}{2}ZX|\psi\rangle \otimes Z|\psi\rangle \otimes (|0\rangle - |1\rangle). \tag{7}$$

Since Z and X anticommute,

$$|\Psi\rangle = \frac{1}{2}X|\psi\rangle \otimes |\psi\rangle \otimes (|0\rangle + |1\rangle) - \frac{1}{2}XZ|\psi\rangle \otimes Z|\psi\rangle \otimes (|0\rangle - |1\rangle). \tag{8}$$

We can simplify this further by realizing that Z_1Z_2 is a stabilizer, so when you act that on the original code state, it just yields the same state with eigenvalue of 1. Therefore,

$$|\Psi\rangle = \frac{1}{2}X|\psi\rangle \otimes |\psi\rangle \otimes (|0\rangle + |1\rangle) - \frac{1}{2}X|\psi\rangle \otimes |\psi\rangle \otimes (|0\rangle - |1\rangle), \tag{9}$$

$$|\Psi\rangle = X|\psi\rangle \otimes |\psi\rangle \otimes |1\rangle. \tag{10}$$

If we measure the ancilla qubit, then we get 1 as well as the original state with the error, $|\Psi\rangle = X|\psi\rangle \otimes |\psi\rangle$! When we apply the same procedure to the second stabilizer, we recover the syndrome $\{1, 0\}$, exactly as expected for an X_1 error, and the quantum state remains unchanged. This syndrome then tells us which corrective gates to apply for correction—so it's crucial that the decoding leaves the state intact.

Going back to the algebra, we notice that the location which determines whether the ancilla measurement will be a 0 to 1 is when we anticommuted the operators. If the operators commuted, there would be no $-$ sign in front of the second term in (8), so the $|1\rangle$ states of the ancilla would cancel, leaving a 0 measurement. This matches up with our definition of an error syndrome perfectly since stabilizers that commute with an error will yield an ancilla measurement of 0 and those that anticommute with an error will yield an ancilla measurement of 1.

Generally speaking, the circuit we used for generating the error syndrome for the simple repetition code can be applied for any stabilizer code, except we would use r ancilla qubits, each corresponding to a controlled stabilizer operator on the code. A general circuit diagram can be found in Figure 2. Then, after the error syndrome is generated, a classical algorithm can perform decoding to determine which error caused the syndrome and the relevant operators can be applied to reverse the error.

III. 5-QUBIT PERFECT STABILIZER CODE

To understand the full power of stabilizer codes, let's take a look at the 5-qubit perfect code. This code is a

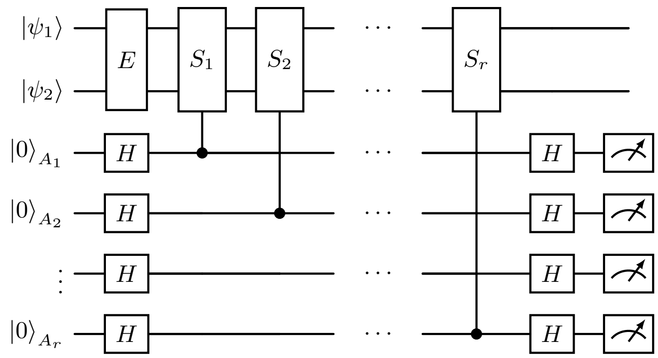


FIG. 2: Decoding Circuit for General Stabilizer Code

[[5, 1, 3]] code, so it takes in 5 physical qubits, encodes 1 qubit, and corrects $(3-1)/2 = 1$ error. In other words, this qubit code is a better version of the simple repetition code, correcting for all single qubit errors. In fact, it was shown that this code has the smallest n for any $k = 1$ and $d = 3$ code [9]. The stabilizer generators for the code are,

$$\begin{gathered} X \otimes Z \otimes Z \otimes X \otimes I \\ I \otimes X \otimes Z \otimes Z \otimes X \\ X \otimes I \otimes X \otimes Z \otimes Z \\ Z \otimes X \otimes I \otimes X \otimes Z \end{gathered}$$

Therefore, the code states are those which are stabilized by the four stabilizers listed above and any linear combination of these stabilizers. Calculating the code words for this code, we get the following logical qubits,

$$\begin{aligned} |0_L\rangle = -\frac{1}{4}\Big(&|00000\rangle + |10010\rangle + |010001\rangle + |10100\rangle \\ &+ |01010\rangle - |11011\rangle - |00110\rangle - |11000\rangle \\ &- |11101\rangle - |00011\rangle - |11110\rangle - |01111\rangle \\ &- |10001\rangle - |01100\rangle - |10111\rangle + |00101\rangle\Big), \end{aligned}$$

$$\begin{aligned} |1_L\rangle = \frac{1}{4}\Big(&|11111\rangle + |01101\rangle + |10110\rangle + |01011\rangle \\ &+ |10101\rangle - |00100\rangle - |11001\rangle - |00111\rangle \\ &- |00010\rangle - |11100\rangle - |00001\rangle - |10000\rangle \\ &- |01110\rangle - |10011\rangle - |01000\rangle + |11010\rangle\Big). \end{aligned}$$

We can now see the usefulness of stabilizer codes. They allow us to concisely write code while also providing us a simple way to detect and correct errors in this code. If we were given the logical states and were asked to correct errors in the code, it would be difficult to figure out what an error would look like on the five qubit code compared to the simple repetition code. However, using the stabilizer generators, we can visualize the effect an error has on the code and correct them accordingly.

Using the syndrome formula defined in (4), we can generate the syndromes of all single qubit errors on the first qubit. To simplify the calculations, we can use the fact that Pauli operators commute with themselves and anti-commute with other Pauli operators.

$E \in \mathcal{G}$	Syndrome
X_1	$\{0,0,0,1\}$
Y_1	$\{1,0,1,1\}$
Z_1	$\{1,0,1,0\}$

TABLE I: Qubit 1 Error Syndromes

As depicted in table I, all the syndromes are non zero, so that means the single qubit errors, on the first qubit, are detected and further corrected by applying the same error on the first qubit.

A. Transversal Operators

Taking a closer look at the code, we notice some patterns in the code. One major pattern is that for each state in $|0\rangle_L$, there is a corresponding state in $|1\rangle_L$ with all the bits flipped. For instance, the second state $|10010\rangle$ in $|0\rangle_L$ appears as $|01101\rangle$ in $|1\rangle_L$. This is reminiscent of the X operators which swaps the 0 and 1 states, so we can propose a logical X operator, $\bar{X}$, to transform between the code states,

$$\begin{aligned} \bar{X}\,|0\rangle_L &= (X \otimes X \otimes X \otimes X \otimes X)\,|0\rangle_L = |1\rangle_L\,, \\ \bar{X}\,|1\rangle_L &= (X \otimes X \otimes X \otimes X \otimes X)\,|1\rangle_L = |0\rangle_L\,. \end{aligned}$$

Additionally, we notice that the number of 1s in each state in $|0\rangle_L$ is even while the number of 1s in each state in $|1\rangle_L$ is odd. This leads us to conclude a logical Z operator, $\bar{Z}$, that operates like a regular Pauli-Z gate on the code space,

$$\begin{aligned} \bar{Z}\,|0\rangle_L &= Z^{\otimes 5}\,|0\rangle_L = (-1)^{2n}\,|0\rangle_L = |0\rangle_L\,, \\ \bar{Z}\,|1\rangle_L &= Z^{\otimes 5}\,|1\rangle_L = (-1)^{2n+1}\,|1\rangle_L = -\,|1\rangle_L\,. \end{aligned}$$

These logical operators are powerful because they allow us to perform qubit operations on the encoded state. A necessary property of the logical $\bar{Z}$ and $\bar{X}$ gates is that they must commute with the stabilizer states [8]. Let M be a stabilizer of the encoded qubit state $|\psi\rangle = a\,|0\rangle_L + b\,|1\rangle_L$,

$$\begin{aligned} \bar{Z}M(a\,|0\rangle_L + b\,|1\rangle_L) &= \bar{Z}(a\,|0\rangle_L + b\,|1\rangle_L) = a\,|0\rangle_L - b\,|1\rangle_L\,, \\ M\bar{Z}(a\,|0\rangle_L + b\,|1\rangle_L) &= M(a\,|0\rangle_L - b\,|1\rangle_L) = a\,|0\rangle_L - b\,|1\rangle_L\,. \end{aligned}$$

So,

$$\bar{Z}M\,|\psi\rangle - M\bar{Z}\,|\psi\rangle = 0 \implies [\bar{Z}, M]\,|\psi\rangle = 0.$$

A similar proof can be done for $\bar{X}$. An intuitive reason as to why $\bar{Z}$ and $\bar{X}$ must commute with all stabilizers is because the logical operators should never remove a state from the code space. If the operators didn't commute with the stabilizers, then $\bar{Z}|\psi\rangle$ or $\bar{X}|\psi\rangle$ could produce a state that isn't a +1 eigenvector of the stabilizer, taking it out of the code space.

Another important property is that $\bar{Z}$ and $\bar{X}$ behave the same as regular Pauli operators relative to each other. In other words, they satisfy a Lie algebra when including the $\bar{Y} = Y^{\otimes 5}$ operator. For the 5-qubit code,

$$[\bar{Z}, \bar{X}] = [Z, X]^{\otimes 5} = (iY)^{\otimes 5} = iY^{\otimes 5} = i\bar{Y},$$
$$[\bar{X}, \bar{Y}] = [X, Y]^{\otimes 5} = (iZ)^{\otimes 5} = iZ^{\otimes 5} = i\bar{Z},$$
$$[\bar{Y}, \bar{Z}] = [Y, Z]^{\otimes 5} = (iX)^{\otimes 5} = iX^{\otimes 5} = i\bar{X}.$$

This property of the logical Pauli operators allows us to abstract away the details of the code itself. Furthermore, we call these operators **transversal operators** since they work by just tensoring identical one qubit gates across each qubit. Transversal operators are key to developing fault tolerant computing because they enable a simple way to perform qubit operators on a quantum code.

Unfortunately, not all operators are transversal. Consider an arbitrary unitary U acting on the stabilizer group S. If a unitary is transversal, it must map any code state back into the code space. This means that $U|\psi\rangle$ must be in the code space. Additionally, let $g \in S$,

$$UgU^\dagger U|\psi\rangle = Ug|\psi\rangle = U|\psi\rangle,$$

implying $UgU^\dagger$ must be a stabilizer of $U|\psi\rangle$. Since $U|\psi\rangle$ is in the code space, $UgU^\dagger$ must also be a stabilizer. In other words, U must transform the stabilizer set back to itself. To see if H is transversal, we need to see what it does to the stabilizer set.

Before we begin the computation, we will first note an important property of the hadamard operator,

$$HZH^\dagger = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$
$$HZH^\dagger = \frac{1}{2}\begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix} = X, \tag{11}$$
$$HXH^\dagger = HHZH^\dagger H^\dagger = Z. \tag{12}$$

where we used the property that H is unitary and $H = H^\dagger \implies H^2 = 1$. So, applying the Hadamard operator to the stabilizer set switches all X operators to Z operators and Z operators to X operators.

If we apply the Hadamard to the first stabilizer generator in the five qubit stabilizer group, we get the operator,

$$M' = ZXXZI.$$

This operator doesn't lie in the group since no combination of products of the generators produces M'. This means that the Hadamard doesn't transform S into S, so it is not transversal.

IV. STABILIZER CODE CHALLENGES

As we have seen, stabilizer codes are powerful quantum codes by enabling an efficient way to detect and correct quantum error. However, there are multiple major bottleneck in using these stabilizer codes. One of the those bottlenecks is the hardness of decoding stabilizer codes. For the simple repetition code, it was trivial what error caused the observed stabilizer syndrome. However, consider the 5-qubit perfect code. Suppose the first qubit has an X error on it. The syndrome for this error would be,

$$[X \otimes I \otimes I \otimes I \otimes I,\ X \otimes Z \otimes Z \otimes X \otimes I] = 0,$$
$$[X \otimes I \otimes I \otimes I \otimes I,\ I \otimes X \otimes Z \otimes Z \otimes X] = 0,$$
$$[X \otimes I \otimes I \otimes I \otimes I,\ X \otimes I \otimes X \otimes Z \otimes Z] = 0,$$
$$\{X \otimes I \otimes I \otimes I \otimes I,\ Z \otimes X \otimes I \otimes X \otimes Z\} = 0.$$

or in other words, $f(E) = \{0, 0, 0, 1\}$ using (4) or equivalently from table I. Since each outcome in the syndrome is a 0 or 1, there are $2^4 = 16$ possible syndromes. Furthermore, there is no obvious pattern with the error and its syndrome, so determining the error given the syndrome is difficult. The $[[5, 1, 3]]$ code is also relatively simple compared to other codes with significantly larger n and k.

Errors can also be degenerate meaning that multiple errors can generate the same error syndrome, adding complexity to syndrome decoding. Consider the error $Z_2Z_3X_4$,

$$[I \otimes Z \otimes Z \otimes X \otimes I, X \otimes Z \otimes Z \otimes X \otimes I] = 0,$$
$$[I \otimes Z \otimes Z \otimes X \otimes I, I \otimes X \otimes Z \otimes Z \otimes X] = 0,$$
$$[I \otimes Z \otimes Z \otimes X \otimes I, X \otimes I \otimes X \otimes Z \otimes Z] = 0,$$
$$\{I \otimes Z \otimes Z \otimes X \otimes I, Z \otimes X \otimes I \otimes X \otimes Z\} = 0,$$
$$f(Z_2Z_3X_4) = \{0, 0, 0, 1\}$$

It has the same error syndrome as X_1! Therefore, while performing syndrome decoding, we can't assume that the error syndromes are bijective. Note that this can also be seen using the pigeonhole principle since we have 16 possible syndromes but $4^5 = 1024$ possible errors.

Multiple papers attempt to devise optimal algorithms for quantum error decoding. Instead of identifying the exact error, these algorithms use probabilistic methods to output the most likely error. One of the most popular algorithms, Degenerate Quantum Maximum Likelihood Decoding (DQMLD) [4], calculates the probability of an error subspace given a syndrome and identifies and outputs the maximum probability error subspace.

However, this algorithm is an **NP-Hard** problem. In fact, the non degenerate version is also in **NP-Complete**, so there haven't been any polynomial algorithms to decode an arbitrary stabilizer syndrome.

Another major challenge facing stabilizer codes is the **No-Go Theorem**:

For any nontrivial local-error-detecting quantum code, the set of transversal, logical unitary operators is not universal [1].

We notice the effects of the No-Go Theorem in the five qubit code. Recall that we showed that the Pauli gates are all transversal; however, the hadamard gate isn't transversal. In order to generate a universal set of logical unitary operators, the hadamard gate is needed in a variety of constructions. Furthermore, for the sets of universal quantum gates which don't include the Hadamard gate, it can be shown that at least one operator won't be transversal for the five qubit code.

A major consequence of this theorem is that it is impossible to find a code with a simple set of logical operators that allow for universal quantum computation. As a result, in order to perform fault-tolerant computation, we must investigate other methods. One promising avenue is the idea of teleporting gates which use qubit measurements combined with transversal gates to generate universal quantum computation [3].

V. FUTURE DIRECTIONS

Despite some setbacks, stabilizer codes are an effective error correcting strategy. In this paper, we covered the stabilizer formulation and how it can be applied to both the simple repetition code and the five qubit perfect code. We also demonstrated the power of stabilizer code to simplify computations by introducing the notion of transversal gates. However, since the introduction of stabilizer codes in 1997 by Gottesman [2], there have been significant advances to the field, exploring new applications for these codes.

One of these applications attempts to solve an issue we discussed in the previous section, decoding. This approach utilizes quantum circuit equivalence rules to simplify circuits. These equivalence rules provide alternate methods to perform the same computation. For instance, as shown in (11) and (12), the simplest equivalence rules are,

$$HZH = X, HXH = Z.$$

Using these rules, it is possible to reduce the size of encoding and decoding circuits [7].

Another area of research dives into a geometric understanding of stabilizer codes. One particular paper uses ZX diagrams [5]. In this graph structure, nodes of the graph can be colored green or red where green nodes represent Z operators and red nodes represent X operators. Edges can also be colored blue to represent hadamard, H, operators.

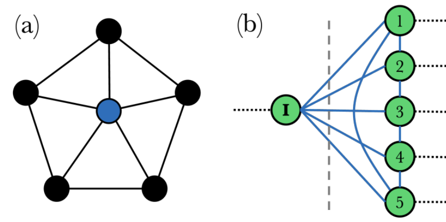


FIG. 3: (a) Graph representation and (b) ZXCF of the 5-qubit code.

Overall, this graph represents a circuit which takes in k qubits and outputs n qubits where the qubits are represented as the end of an edge without a node. These representations are valuable in understanding the structure of quantum codes. The five qubit code, for example, is represented in Figure 3.

By starting from a graph and working backwards, the hope is to generate new stabilizer codes with unique properties related to the graph structure.

[1] Bryan Eastin and Emanuel Knill. Restrictions on transversal encoded quantum gate sets. *Phys. Rev. Lett.*, 102:110502, Mar 2009.

[2] Daniel Gottesman. Stabilizer codes and quantum error correction, 1997.

[3] Daniel Gottesman and Isaac L. Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations.

[4] Pavithran Iyer and David Poulin. Hardness of decoding quantum stabilizer codes, 2013.

[5] Andrey Boris Khesin, Jonathan Z. Lu, and Peter W. Shor. Universal graph representation of stabilizer codes, 2025.

[6] Zhe Meng, Wen-Qiang Liu, Bo-Wen Song, Xiao-Yun Wang, An-Ning Zhang, and Zhang-Qi Yin. Experimental realization of universal high-dimensional quantum gates with ultra-high fidelity and efficiency, 2023.

[7] Arijit Mondal and Keshab K. Parhi. Optimization of quantum circuits for stabilizer codes. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71(8):3647–3657, 2024.

[8] Joschka Roffe. Quantum error correction: an introductory guide. *Contemporary Physics*, 60(3):226–245, July 2019.

[9] Sixia Yu, Jürgen Bierbrauer, Ying Dong, Qing Chen, and C. H. Oh. All the stabilizer codes of distance 3. *IEEE Transactions on Information Theory*, 59(8):5179–5185, 2013.