CSCI 581:
Quantum Error Correction: From Classical Codes to the LDPC Frontiers

Om Biyani, Tristan Yan

April

Introduction

Quantum computers are inherently fragile, with noise and decoherence posing major challenges to scalability. To protect quantum information, quantum error-correcting codes (QECCs) are employed to encode logical qubits into entangled states across multiple physical qubits.

This report traces the development of QECCs from classical linear codes to modern quantum constructions. We begin by reviewing classical coding theory and the role of distance in error correction. We then introduce stabilizer codes, focusing on the $[[5, 1, 3]]$ perfect code, which we implement in Qiskit to demonstrate syndrome extraction and recovery.

From there, we explore quantum low-density parity-check (LDPC) codes — an important class of sparse stabilizer codes—and the ongoing search for "good" LDPC codes with scalable distance and logical capacity. Finally, we examine toric codes, a key example of quantum LDPC codes that, while not "good" in the scaling sense, represent an important step toward the design of scalable, topologically structured quantum error-correcting codes.

Classical Linear Block Codes

Quantum error correction builds on foundational ideas from classical coding theory, particularly linear block codes. An (n, k) linear block code encodes k bits of logical information into $n > k$ physical bits. The set of valid codewords forms a k-dimensional subspace $C \subseteq \{0, 1\}^n$ [4].

A generator matrix $G \in \{0, 1\}^{k \times n}$ has linearly independent rows that span this subspace. A message $u \in \{0, 1\}^k$ is encoded by computing codeword $c = u \cdot G$, resulting in a valid codeword $c \in C$.

To check whether a received codeword has been corrupted due to an error, we use a parity-check matrix $H \in \{0, 1\}^{(n-k)\times n}$, whose rows span the orthogonal complement $C^\perp$. A word c is valid if and only if:

$$Hc^T = 0$$

If this condition fails, the result Hc^T is called the **syndrome**, and it indicates the presence and structure of an error.

Low-Density Parity-Check (LDPC) Codes

A **low-density parity-check** (LDPC) code is a linear binary block code whose parity-check matrix has a low density of ones [4]. This sparsity reduces the complexity of decoding and makes LDPC codes highly scalable in classical applications. These ideas extend naturally into quantum error correction, where sparse stabilizer structures are constructed in a similar fashion.

Code Distance and Error Correction Thresholds

A fundamental question in error correction is: how many errors can a code detect and correct? The answer depends on the code's **distance** d (also referred to as the minimum Hamming distance $d_{\min}$).

In classical coding theory, the **Hamming distance** between two codewords is the number of bit positions in which they differ. For a linear block code, the code distance d is defined as the minimum Hamming distance between any two valid codewords. A classical $[n, k, d]$ code can detect up to $d - 1$ errors and correct up to $\lfloor \frac{d-1}{2} \rfloor$ bit errors [7].

This notion extends naturally to quantum error correction. A quantum code of distance d can detect any error acting nontrivially on fewer than d qubits and correct up to $\lfloor \frac{d-1}{2} \rfloor$ arbitrary single-qubit errors [10].

In the quantum setting, code distance is defined as the minimum number of qubits on which a nontrivial logical operator must act to map one valid codeword to another. The larger the distance, the more errors a code can tolerate before logical information is compromised. For this reason, d is often referred to as the code's **correction power**.

The stabilizer formalism provides a powerful framework for defining quantum codes and analyzing their distance, structure, and correction capabilities. We explore this next.

Stabilizer Quantum Codes

Quantum stabilizer codes generalize the concept of classical linear block codes. A $[[n, k, d]]$ stabilizer quantum code encodes k logical qubits into a subspace of the n-qubit Hilbert space $(\mathbb{C}^2)^{\otimes n}$, defined as the $+1$ eigenspace of an abelian subgroup S of the n-qubit Pauli group [6].

The Pauli group on n qubits is given by:

$$\mathcal{P}_n = \{\lambda\, P_1 \otimes P_2 \otimes \cdots \otimes P_n \,|\, P_i \in \{I, X, Y, Z\},\ \lambda \in \{\pm 1, \pm i\}\}$$

The stabilizer group $S \subset \mathcal{P}_n$ contains $n - k$ independent, commuting generators (called stabilizer checks or stabilizer generators), and excludes any element proportional to $-I$. This ensures that all valid codewords $|\psi\rangle$ satisfy:

$$g|\psi\rangle = |\psi\rangle \quad \text{for all } g \in S$$

In other words, the code space is **stabilized** by all elements of S. Any state outside this space—i.e., an invalid codeword—will lie in a different eigenspace and be detectable via measurement [6].

Syndrome Measurement

Just as classical codes use parity-check matrices to detect errors by computing a syndrome Hc^T, stabilizer codes detect quantum errors by measuring the stabilizer generators.

Each generator $g_i \in S$ is a Pauli operator with eigenvalues ± 1. Measuring g_i on a quantum state yields either $+1$ (the state is in the $+1$ eigenspace of g_i) or -1 (the state has been perturbed by an error that anticommutes with g_i). The full syndrome is a binary string of $n - k$ bits, where each bit reflects the outcome of measuring a corresponding generator.

Because the stabilizer generators commute with one another and with all logical operators, these measurements do not collapse the logical state. Instead, the syndrome reveals how the error operator commutes or anticommutes with the stabilizer group, allowing the decoder to identify the most likely error and apply a correction—restoring the encoded state without directly measuring it [10].

Case Study: The $[[5, 1, 3]]$ Perfect Code

To understand the principles of stabilizer quantum error correction, we implemented the $[[5, 1, 3]]$ code, also known as the five-qubit or "perfect" code. It is the smallest possible quantum code capable of correcting an arbitrary single-qubit error, with parameters $[[n = 5, k = 1, d = 3]]$. This code encodes one logical qubit into five physical qubits and can correct any single-qubit Pauli error ($t = \lfloor \frac{d-1}{2} \rfloor = 1$) [9].

Our implementation follows the logical state definitions and stabilizer structure presented in [12], and is constructed using Qiskit. We explicitly define the logical basis states, apply stabilizer measurements, and simulate all possible single-qubit Pauli errors. We generate a complete syndrome table that demonstrates the code's ability to identify each correctable error.

The stabilizer generators for the $[[5, 1, 3]]$ code are:

$$\begin{aligned} S_0 &= IZXXZ \\ S_1 &= XXZIZ \\ S_2 &= XZIZX \\ S_3 &= ZIZXX \end{aligned}$$

The encodings for the logical basis states $|0_L\rangle$ and $|1_L\rangle$ are defined as follows:

$$\begin{aligned} |0_L\rangle = \frac{1}{4}\big(\,&|00000\rangle + |11000\rangle + |01100\rangle + |00110\rangle \\ &+ |00011\rangle + |10001\rangle - |10100\rangle - |01010\rangle \\ &- |00101\rangle - |10010\rangle - |01001\rangle - |11110\rangle \\ &- |01111\rangle - |10111\rangle - |11011\rangle - |11101\rangle\big) \end{aligned}$$

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

These logical basis states lie in the $+1$ eigenspace of all four stabilizer generators. Their construction from the stabilizer group involves satisfying specific eigenvalue constraints, which we do not detail here; the idea behind the derivation is outlined in [9]

To test the error-correcting capacity of this code, we simulate all possible single-qubit Pauli errors and measure the resulting stabilizer syndromes. The results are summarized in Table 1, where each error is mapped to a unique 4-bit syndrome.

Each unique syndrome corresponds to a specific single-qubit Pauli error. For any single-qubit error, the decoder identifies the corresponding syndrome and applies the appropriate correction. For example, if the measured syndrome is 1010, this indicates an X error on qubit 2, and applying another X gate to that qubit completes the recovery. This works because all single-qubit Pauli operators are self-inverse.

Error Type	Qubit	Syndrome
X	0	1000
X	1	0101
X	2	1010
X	3	0100
X	4	0011
Z	0	0110
Z	1	0010
Z	2	0001
Z	3	1001
Z	4	1100
Y	0	1110
Y	1	0111
Y	2	1011
Y	3	1101
Y	4	1111

Table 1: Syndromes for all single-qubit Pauli errors. Each row lists the type of error, the affected qubit (indexed 0–4), and the resulting 4-bit syndrome.

By implementing the $[[5, 1, 3]]$ code, we demonstrated how stabilizer codes work in practice—how logical qubits can be encoded, how errors manifest as measurable syndromes, and how recovery restores the original state. This implementation lays the groundwork for exploring more sophisticated quantum error-correcting codes.

The Quantum Hamming Bound

In both classical and quantum coding theory, the Hamming bound places a limit on the number of errors a code can reliably detect and correct, given a fixed number of physical bits or qubits.

In the classical case, the Hamming bound for a code that corrects up to t errors is [7]:

$$\sum_{i=0}^{t} \binom{n}{i} \leq 2^{n-k}$$

This inequality ensures that the total number of correctable errors does not exceed the number of distinct syndrome patterns available in the code space.

In the quantum setting, the bound becomes stricter due to more types of quantum errors. A quantum error-correcting code $[[n, k, d]]$ that can correct all errors affecting up to $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ qubits must satisfy [10]:

$$\sum_{i=0}^{t} 3^i \binom{n}{i} \leq 2^{n-k}$$

The factor of 3^i accounts for the three possible nontrivial Pauli errors (X, Y, and Z) that can affect each qubit. Notably, the $[[5, 1, 3]]$ code saturates the quantum hamming bound, that's why it's called the perfect code.

The quantum Hamming bound imposes a strict trade-off between distance, rate, and code length—boosting error-correction capability by increasing distance d typically reduces the code rate k/n. As a result, constructing quantum codes that are both efficient and robust is difficult. In practice, rather than optimizing for theoretical limits, researchers prioritize codes that are sparse, scalable, and hardware-efficient. This motivates the study of structured families like quantum LDPC codes, which, while not saturating the bound, offer a practical path toward fault-tolerant quantum computing.

Calderbank-Shor-Steane Codes

A Calderbank-Shor-Steane (CSS) Code is a pair of classical binary linear block codes C_X, $C_Z \subset \mathbb{F}_2^n$ (n-dimensional Binary field) such that $C_X \subset C_Z^{\perp}$ [3]. C_X and C_Z are defined by their parity check matrices H_X and H_Z. As a result of $C_X \subset C_Z^{\perp}$, $H_Z H_X^T = 0 \mod 2$.

A CSS code defines a stabilizer code where the stabilizer group is generated by X_c defined as

$$X_c = \bigotimes_{i=1}^{n} X_i^{c_i}.$$

c is a row of the matrix H_X, and $X_i^{c_i}$ means applying the Pauli X-operator (bit-flip) to the i-th qubit according to the c_i entry of the row.

The stabilizer check Z_d is similarly defined as

$$Z_d = \bigotimes_{i=1}^{n} Z_i^{d_i},$$

where d is a row of the matrix H_Z, and $Z_i^{d_i}$ applies the Pauli Z-operator (phase-flip) to the i-th qubit according to the d_i entry.

Shown by Kitaev, any $[[n, k, d]]$ stabilizer code can be mapped onto a $[[4n, 2k, 2d]]$ CSS code [1, 8].

Quantum LDPC

A quantum LDPC code is a family of stabilizer codes such that the number of qubits participating in each check operator and the number of stabilizer checks in which each qubit participates are bounded by a constant.

In terms of a CSS code this means the number of 1 in each row and column of H_X and H_Z is bounded by a constant. Use of LDPC reduces the complexity of decoding coding as smaller quantity of qubits participate in stabilizer checks.

Good LDPC Codes

A major open problem in quantum coding theory is whether "good" LDPC codes exist—codes where both the number of logical qubits k and the code distance d scale linearly with the total number of qubits n, i.e., $k, d \in \mathcal{O}(n)$.

Most known quantum LDPC codes achieve only sublinear distance, typically $d = \Omega(\sqrt{n}\,\text{polylog}(n))$ [5]. Recent constructions based on homological methods, high-dimensional expanders, and tensor products aim to close this gap, but fully good quantum LDPC codes remain elusive.

Tanner Graphs

Any linear binary code can be represented as a Tanner Graph. A Tanner Graph is a type of bipartite graph, a graph where the nodes can be partitioned into two classes, and no edge connects two nodes from the same class [4]. In the Quantum Stabilizer case there is a n nodes for each physical qubit and $n-k$ nodes for each parity check. Edges connect the parity checks to qubits if and only if that qubit is involved in that parity check.

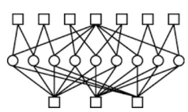

Figure 1: Tanner graph of shores code. Top squares are Pauli-X stablizer checks, circles are qubits, and bottom squares are Pauli-Z stablizer checks [3]

Tanner Graphs are a commonly used tool to visualize error correction using CSS and similar stabilizer based error correction methods.

Introduction to Homology

Homology is a concept in algebraic topology that uses the structure of chain complexes to study the topology of spaces. Chain complexes are $n+1$ collections of $\mathbb{F}_2$ vector spaces, C_i, and linear maps, ∂_i called boundary maps which describe the relationships between different dimensional components of an algebraic object [3].

$$C = (C_n \xrightarrow{\partial_n} C_{n-1} \xrightarrow{\partial_{n-1}} \cdots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0)$$

Chain complexes must fulfill the Chain Complex Condition

$$\partial_i \circ \partial_{i+1} = 0 \quad \text{for all } i.$$

One of the derivation methods for LDPC codes comes from analysis of homological structures.

We will define a few terms that are helpful for this understanding of this section. i-cells are basis vectors of C_i; i-chains are Elements of C_i; i-cycles are elements of a chain space, C_i, that lie in the kernel of a boundary map ∂_i; i-cycle is an element of C_i in the image of ∂_{i+1}, and the i-th homology is defined $H_i(X) = \frac{\ker\, \partial_{i+1}}{\text{img}\, \partial_i}$ [3, 8].

Homology In Error Correction

Homological Linear block Codes

We can represent (n, k) linear block codes using this homological structure.

Let $C_1 \xrightarrow{\partial_1} C_0$ be a chain complex, where: n physical bits are represented as i-chains in C_1, k encoded logical bits are i-chains in C_0, ∂_1 corresponds to the parity-check matrix H, i-cycles of ∂_1 are valid codewords, and i-boundaries correspond to errors in the code.

Homological CSS Codes

Similarly, $[[n, k, d]]$ CSS codes can be represented in homological structure [3, 8].

Let $C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0$ be a chain complex, where: n physical qubits are represented as i-chains in C_1, ∂_2 corresponds to the stabilizer-checks H_Z^T, ∂_1 corresponds to the stabilizer-checks H_X, by definition CSS codes fulfill the Chain complex rule $H_Z H_X^T = 0 \mod 2$, i-cycles of ∂_2 and ∂_1 are valid codewords, and i-boundaries correspond to errors in the code.

Toric Codes

A promising source of "good" LDPC codes comes from homology. By tessellating homological structures, a class of LDPC codes known as Topological Codes has been developed. The simplest example of this is the Toric Code, which is an LDPC method derived from the tessellation of squares on the surface of a torus [2, 3].

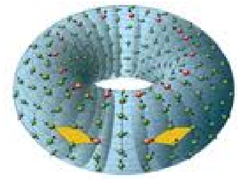

Figure 2: Tessellated Torus [2]

Within each square on the tessellated torus a chain complex is defined $C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0$. For a single square the tanner graphs is given as:

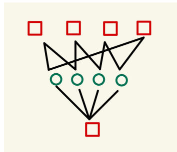

Figure 3: Tanner Graph for Single Square

The vertices of the square represent the physical qubits, the edges of the square correspond to Pauli-X parity checks, and the face of the square represents the Pauli-Z parity check. ∂_2, the Pauli-X parity check matrix, is encoded as the boundary map:

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

The first row represents the first edge, connecting the first and second vertices (qubits), and the second edge connects the second and third vertices. This scheme fills the boundary map for all of the edges on the square. Anywhere there is a '1' in the matrix, it indicates an X-stabilizer check, and a Pauli-X would be applied to each qubit with a '1' during the stabilizer check.

The Pauli-Z parity check, ∂_2 is defined by the boundary map:

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

The face of the square is formed by all four qubits, meaning that all qubits participate in the Z-stabilizer check.

Two logical qubits are encoded in each square, and two-qubit errors must occur to change the logical state, making a single square Toric Code a [[4,2,2]] code. This scheme can be extended through tessellation of the entire torus, increasing the resilience of logical qubits. The Toric Code is the simplest example of topology-based error correction.

Other Topological Codes

The search for effective LDPC codes continues through this topology-based error correction scheme. Codes derived from topological methods are part of a broader family known as surface codes. These LDPC codes are constructed through various forms of tessellation across the surface of homological objects [3].

This class of codes is too vast and complex to fully discuss within the scope of this paper, but a few well-known surface codes include the Shor code, which is a tessellation of the real projective plane [3, 11], and hyperbolic surface codes, which are more intricate as their tessellations are derived from hyperbolic geometry [3]. There are several other codes, and for more information, you can refer to [1].

Conclusions

Quantum error correction is essential for building scalable, fault-tolerant quantum computers. In this report, we traced the evolution of quantum codes from classical linear block codes to stabilizer codes, and implemented the $[[5, 1, 3]]$ perfect code in Qiskit to demonstrate syndrome measurement and error recovery in practice.

We then explored structured families like CSS and LDPC codes, which aim to balance error-correction performance with hardware efficiency. While fully "good" quantum LDPC codes remain an open challenge, progress in homological and topological constructions—such as toric codes—offers promising pathways forward.

Our exploration covers just a small part of the broader landscape, but it provides a solid foundation for understanding how quantum codes are built and why they matter.

References

[1] Qubit css code. In Victor V. Albert and Philippe Faist, editors, *The Error Correction Zoo*. 2024.

[2] H. Bombin, Ruben S. Andrist, Masayuki Ohzeki, Helmut G. Katzgraber, and M. A. Martin-Delgado. Strong resilience of topological codes to depolarization. *Phys. Rev. X*, 2:021004, Apr 2012.

[3] Nikolas P. Breuckmann and Jens Niklas Eberhardt. Quantum low-density parity-check codes. *PRX Quantum*, 2:040101, Oct 2021.

[4] Jr. Daniel J. Costello. An introduction to low-density parity check codes. Technical report, Department of Electrical Engineering, University of Notre Dame, August 2009. The author gratefully acknowledges the help of Tom Fuja.

[5] Michael H. Freedman, David A. Meyer, and Feng Luo. Z2-systolic freedom and quantum codes. In Ranee K. Brylinski and Goong Chen, editors, *Mathematics of Quantum Computation*, pages 287–320. Chapman & Hall/CRC, 2002.

[6] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD thesis, California Institute of Technology, 1997.

[7] Richard W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[8] Alexei Kitaev. Anyons in an exactly solved model and beyond. *Annals of Physics*, 321(1):2–111, January 2006.

[9] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech Hubert Zurek. Perfect quantum error correcting code. *Physical Review Letters*, 77(1):198–201, 1996.

[10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[11] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

[12] Dave Touchette, Haleemur Ali, and Michael Hilke. 5-qubit quantum error correction in a charge qubit quantum computer. *arXiv preprint arXiv:1010.3242*, 2010.