An Introduction to Quantum Error Correction

Nicholas Connolly

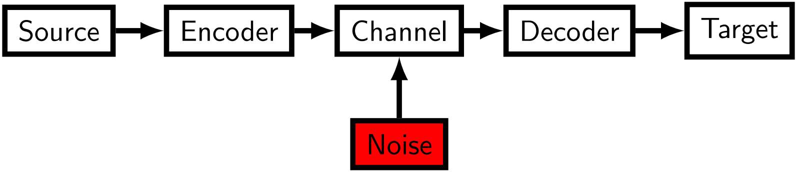
INRIA

November 4, 2022

Table of Contents

Coding Theory

Coding theory [4] is concerned with the study of how to **encode information** for transmission through a **communication channel**.



Physical communication channels are always subject to the presence of **noise** which can disrupt the information.

Classical Bits and Bit-flip Errors

In a **classical** setting, information is represented using strings of **bits**, each of which has value 0 or 1.

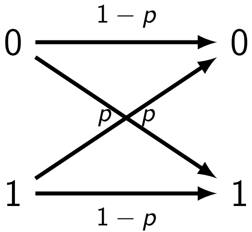
010011000111

A noisy channel can lead to **bit-flip errors** during transmission, where the value of a bit changes from 0 to 1 or from 1 to 0.

010**1**11000**1**0**1**

Binary Symmetric Channel

A **binary symmetric channel** is a noisy channel in which the value of each transmitted bit can be flipped with probability $0 \leq p \leq 1$.



By encoding the information using **error correcting codes**, many mistakes in a transmitted message can be identified and fixed.

Linear Codes

A (binary) **linear code** C is a vector subspace of $\mathbb{F}_2^n$.

1. The vectors in C are referred to as **code words**.
2. C has **length** n, the number of bits in a code word.
3. C has **dimension** k as a subspace of $\mathbb{F}_2^n$.

Example: A code of length 3 and dimension 2 as a subspace of $\mathbb{F}_2^3$

$$\mathbb{F}_2^3 = \left\{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\}$$

$$C = \langle \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix} \rangle = \left\{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix} \right\} = \ker(\begin{bmatrix}0 & 0 & 1\end{bmatrix})$$

Parity Check Matrices

Equivalently, a linear code C may be defined as the *kernel* of a **parity check matrix** H. In other words, $C = \ker(H)$.

$$v \in C \text{ if and only if } Hv = 0 \text{ (modulo 2)}$$

Example: 3-bit Repetition Code

$$H = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \qquad C = \ker(H) = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}$$

For example, $v = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \notin C$ since $Hv = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \neq 0$.

(All bits in a vector of the repetition code have the same value.)

Error Detection

The repetition code is an example of an **error detecting** code.

$$\begin{array}{ccccc} \text{message sent} & \Rightarrow & \text{noisy channel} & \Rightarrow & \text{message received} \\ x & & x+e & & x' \end{array}$$

$$\begin{bmatrix}1\\1\\1\end{bmatrix} \qquad \begin{bmatrix}1\\1\\1\end{bmatrix} + \begin{bmatrix}0\\0\\1\end{bmatrix} \qquad \begin{bmatrix}1\\1\\0\end{bmatrix}$$

Noise can be represented by adding an **error vector** e to the original message x. If $Hx' \neq 0$, then we know there is an error.

$$Hx' \;=\; H(x+e) \;=\; Hx + He \;=\; He$$

Note: If $He = 0$, then e is an undetectable error!

Error Correction and Syndrome Analysis

$$Hx' \;=\; Hx + He \;=\; He \;=\; s$$

The value $He = s$ is known as the **syndrome** of an error e.
The goal of syndrome analysis is to compute e given s.
This allows the original message to be recovered by canceling out e.

$$x' + e \;=\; x + e + e \;=\; x$$

The linear algebra problem $He = s$ can be solved, for example, using Gaussian elimination.

In general, there exists more than one error satisfying $He = s$, but we assume that $|e|$ is small ("maximum likelihood decoding").

Minimum Distance

The **Hamming weight** of a binary vector is the number of nonzero entries.

$$w\left(\begin{bmatrix}1 & 1 & 0\end{bmatrix}^T\right) = 2$$

The **distance** between two vectors is the weight of their sum.

$$d\left(\begin{bmatrix}1 & 1 & 0\end{bmatrix}^T, \begin{bmatrix}0 & 1 & 1\end{bmatrix}^T\right) = w\left(\begin{bmatrix}1 & 0 & 1\end{bmatrix}^T\right) = 2$$

The **minimum distance** of a code C is defined to be the smallest possible distance between any two code words in C.

$$\text{min dist}(C) = \min\{d(v_1, v_2) : v_1, v_2 \in C\}$$

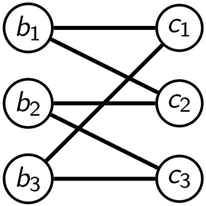
A binary code with length n, dimension k, and minimum distance d is denoted as an $[n, k, d]_2$ code.

Example: The 3-bit repetition code is a $[3, 1, 3]_2$ code.

Tanner Graphs

The **Tanner graph** of a linear code $C = \ker(H)$ is the *bipartite graph* G defined using H as an *incidence matrix*.

$G =$



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

- The columns of H correspond to *bit nodes* in G.
- The rows of H correspond to *check nodes* in G.
- There exists an edge between bit i and check j in G if and only if the entry in cell i, j of H is nonzero.

Table of Contents

Qubits

Whereas classical information is represented using bits, **quantum information** is represented using **qubits** ("quantum bits").

A classical bit has only two possible states: 0 or 1.
A qubit $|\psi\rangle$ can be represented as a *superposition* of states.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad \alpha, \beta \in \mathbb{C} \text{ where } |\alpha|^2 + |\beta|^2 = 1$$

A qubit $|\psi\rangle$ can be thought of as a unit vector in a 2-dimensional complex vector space with a basis formed by the states $|0\rangle$ and $|1\rangle$.

Example: $|+\rangle$ and $|-\rangle$ states

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Quantum Gates

Using the two computational basis states $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, a quantum state $|\psi\rangle$ can be represented as a linear combination.

$$|\psi\rangle \quad = \quad \alpha|0\rangle + \beta|1\rangle \quad = \quad \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad = \quad \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

Similar to logic gates for classical bits, **quantum gates** can be used to describe how to move one quantum state to another. Quantum gates on a single qubit can be represented using 2×2 matrices.

Example: Quantum NOT Gate: $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

$$X|\psi\rangle \quad = \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad = \quad \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \quad = \quad \beta|0\rangle + \alpha|1\rangle$$

Weird Difficulties with Quantum Information

Qubits behave very differently from classical bits.

1. **No cloning theorem**
   Quantum states cannot be duplicated.
2. **Continuous errors**
   A continuum of errors may occur on a single qubit.
3. **Measurements destroy quantum information**
   Measuring a qubit in state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ collapses the state to either 0 or 1 (with probabilities $|\alpha|^2$ and $|\beta|^2$).

Quantum error correction must work around these constraints.

Pauli Operators

A quantum gate U acting on a state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ must be *unitary* ($U^\dagger U = I$) to preserve the condition that $|\alpha|^2 + |\beta|^2 = 1$.

Four important quantum gates are known as the **Pauli Operators**.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Any transformation of a quantum state (including noise) can be described using *only* linear combinations of Pauli operators .

Example: Hadamard Gate H

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} X + \frac{1}{\sqrt{2}} Z$$

Quantum Bit-Flips and Phase-Flips

The I Pauli operator is identity.

$$I|\psi\rangle = |\psi\rangle$$

The X Pauli operator describes a quantum *bit-flip*.

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

The Z Pauli operator describes a quantum *phase-flip*.

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

The Y Pauli operator combines both a bit-flip and a phase-flip.

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = i \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = iXZ$$

Discretization of Quantum Errors

The effect of noise in a quantum channel can be interpreted as a transformation of a quantum state described by an operator E.

$$|\psi\rangle \quad \xrightarrow{\text{noise}} \quad E|\psi\rangle$$

E can be expanded as a linear combination of the Pauli operators.

$$E = e_0 I + e_1 X + e_2 Z + e_3 XZ$$

In the single qubit case, the quantum state $E|\psi\rangle$ is a superposition of four possible terms. These can be exploited for error correction.

$$E|\psi\rangle = e_0|\psi\rangle + e_1 X|\psi\rangle + e_2 Z|\psi\rangle + e_3 XZ|\psi\rangle$$

Measuring the **syndrome** of $E|\psi\rangle$ *collapses* the state onto one of these four possibilities *and* tells us which correction to apply!

Multiple Qubits

A state in a two qubit system, for example, can be expressed as:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle.$$

This notation is a shorthand for *tensor products* of single qubit states. More generally, a multiple qubit state has the form:

$$|0000\rangle = |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle.$$

Tensor products of Pauli operators can be applied linearly.

$$\begin{aligned} X_1 X_3|0000\rangle &= (X \otimes I \otimes X \otimes I)(|0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle) \\ &= X|0\rangle \otimes I|0\rangle \otimes X|0\rangle \otimes I|0\rangle \\ &= |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \\ &= |1010\rangle \end{aligned}$$

Table of Contents

The Three Qubit Bit-Flip Code

Consider a 3-qubit quantum code with basis states $\{|000\rangle, |111\rangle\}$, with states superpositions of the form $|\psi\rangle = \alpha|000\rangle + \beta|111\rangle$.
A bit-flip (X Pauli error) occurring on the first qubit has the form:

$$X_1|\psi\rangle = \alpha X_1|000\rangle + \beta X_1|111\rangle = \alpha|100\rangle + \beta|011\rangle.$$

A bit-flip on at most one qubit is detected using operators Z_iZ_j.

$$Z_iZ_j|\psi\rangle = \begin{cases} |\psi\rangle & \text{if qubit } i \text{ and } j \text{ have the same value} \\ -|\psi\rangle & \text{otherwise} \end{cases}$$

Note that Z_iZ_j leaves states $|\psi\rangle = \alpha|000\rangle + \beta|111\rangle$ fixed.
The syndrome measurement of an error $E|\psi\rangle$ is computed as:

$$\langle\psi|E^\dagger Z_iZ_jE|\psi\rangle = \begin{cases} \langle\psi|E^\dagger E|\psi\rangle = 1 & \text{if } Z_iZ_jE|\psi\rangle = E|\psi\rangle \\ -\langle\psi|E^\dagger E|\psi\rangle = -1 & \text{if } Z_iZ_jE|\psi\rangle = -E|\psi\rangle \end{cases}.$$

Syndrome Analysis of the Three Qubit Bit-Flip Code

The effect of noise on a state $|\psi\rangle$ is modeled by an operator E.

$$|\psi\rangle = \alpha|000\rangle + \beta|111\rangle \qquad \xrightarrow{\text{noise}} \qquad E|\psi\rangle$$

If $E \in \{I, X_1, X_2, X_3\}$, this can be inferred by analyzing the syndrome measurements of the operators $Z_1 Z_2$ and $Z_2 Z_3$.

E	$\langle\psi\|E^\dagger Z_1 Z_2 E\|\psi\rangle$	$\langle\psi\|E^\dagger Z_2 Z_3 E\|\psi\rangle$
I	$+1$	$+1$
X_1	-1	$+1$
X_2	-1	-1
X_3	$+1$	-1

Syndrome measurement does *not* reveal information about α and β, only about E. A correction to $E|\psi\rangle$ is applied based on this.

Pauli Group on n Qubits

The **Pauli group** on 1 qubit is the multiplicative matrix group

$$G_1 = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}.$$

The Pauli matrices X, Y, and Z *anti-commute* with each other.

$$XY = -YX \qquad XZ = -ZX \qquad YZ = -ZY$$

More generally, the Pauli group G_n on n qubits is the group of all n-fold tensor products of Pauli matrices (with factors ± 1 and $\pm i$).

Example:

$$\begin{aligned} Z \otimes Z \otimes I &= Z_1 Z_2 \in G_3 \\ I \otimes Z \otimes Z &= Z_2 Z_3 \in G_3 \end{aligned}$$

Stabilizer Codes

For any subgroup $S \subseteq G_n$, define V_S to be the vector space of n qubit states which are **stabilized** (left fixed) by the elements in S.

$$V_S = \{|\psi\rangle : U|\psi\rangle = |\psi\rangle \text{ for all } U \in S\}$$

Example: Three Qubit Bit-Flip Code

$$S = \langle Z_1 Z_2, Z_2 Z_3 \rangle = \{I, Z_1 Z_2, Z_2 Z_3, Z_1 Z_3\} \subseteq G_3$$

$$V_S = \langle |000\rangle, |111\rangle \rangle = \{|\psi\rangle = \alpha|000\rangle + \beta|111\rangle : |\alpha|^2 + |\beta|^2 = 1\}$$

- $V_S = \langle |000\rangle, |111\rangle \rangle$ is a **stabilizer code**.
- $S = \langle Z_1 Z_2, Z_2 Z_3 \rangle$ is the **stabilizer subgroup**.
- $Z_1 Z_2$ and $Z_2 Z_3$ are the **stabilizer generators**.

V_S is nontrivial if and only if S is commutative and $-I \notin S$.

The Steane Code

The **Steane Code** [8] is an example of a 7-qubit stabilizer code.
It can correct arbitrary errors on a single qubit.

$$\begin{aligned} g_1 &= X_4X_5X_6X_7 \\ g_2 &= X_2X_3X_6X_7 \\ g_3 &= X_1X_3X_5X_7 \\ g_4 &= Z_4Z_5Z_6Z_7 \\ g_5 &= Z_2Z_3Z_6Z_7 \\ g_6 &= Z_1Z_3Z_5Z_7 \end{aligned}$$

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

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

It is defined via the stabilizer generators $\langle g_1, g_2, g_3, g_4, g_5, g_6 \rangle$.
It is also an example of a CSS code using the above two matrices.

Calderbank-Shor-Steane (CSS) Codes

A **CSS** code [1, 8] $CSS(C_X, C_Z)$ is a quantum code defined by two classical linear codes $C_X = \ker(H_X)$ and $C_Z = \ker(H_Z)$.

- $CSS(C_X, C_Z)$ is a special case of stabilizer code.
- The rows of H_X define X-Pauli stabilizer generators: $X_i X_j \cdots$.
- The rows of H_Z define Z-Pauli stabilizer generators: $Z_r Z_s \cdots$.

To be a stabilizer code, the X- and Z-stabilizers *must commute*. For $CSS(C_X, C_Z)$, this requirement is equivalent to the conditions:

- the rows of H_X and rows of H_Z are orthogonal;
- $H_X H_Z^T = H_Z H_X^T = 0$;
- the dual codes satisfy $C_X^\perp \subseteq C_Z$ and $C_Z^\perp \subseteq C_X$.

Note: H_X detects Z-Pauli errors; H_Z detects X-Pauli errors.

Syndrome Analysis for CSS Codes

Similar to how X- and Z-stabilizers are represented using vectors, we can give an analogous description to an error operator $E \in G_n$.

$$E \;=\; X_1 Z_1 X_2 Z_4 \;\in\; G_4 \qquad \Leftrightarrow \qquad \begin{aligned} e_X &= \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T \\ e_Z &= \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^T \end{aligned}$$

Syndrome analysis for the quantum code $CSS(C_X, C_Z)$ can be performed using the classical parity check matrices H_X and H_Z.

- Syndrome for X-type Pauli errors: $H_Z e_X = s_X$
- Syndrome for Z-type Pauli errors: $H_X e_Z = s_Z$

Predicting the quantum error operator E then reduces to the classical problem of predicting e_X and e_Z given s_X and s_Z.

The Shor Code [7]

The **Shor Code** is another example of a CSS code, with 9 qubits. It is a combination of the 3-qubit Bit-Flip and Phase-Flip codes.

$$
\begin{aligned}
g_1 &= Z_1 Z_2 \\
g_2 &= Z_2 Z_3 \\
g_3 &= Z_4 Z_5 \\
g_4 &= Z_5 Z_6 \\
g_5 &= Z_7 Z_8 \\
g_6 &= Z_8 Z_9 \\
g_7 &= X_1 X_2 X_3 X_4 X_5 X_6 \\
g_8 &= X_4 X_5 X_6 X_7 X_8 X_9
\end{aligned}
$$

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

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

Like the Steane code, it can correct any errors on a single qubit.

The 5-Qubit Code

The 5-qubit code is an example of a stabilizer code which is *not* CSS since the stabilizers cannot be separated into X- and Z-types.

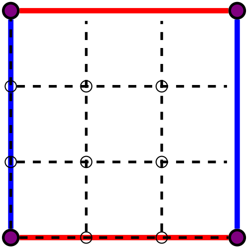
$$
\begin{aligned}
g_1 &= X_1 Z_2 Z_3 X_4 \\
g_2 &= X_2 Z_3 Z_4 X_5 \\
g_3 &= X_1 X_3 Z_4 Z_5 \\
g_4 &= Z_1 X_2 X_4 Z_5
\end{aligned}
$$

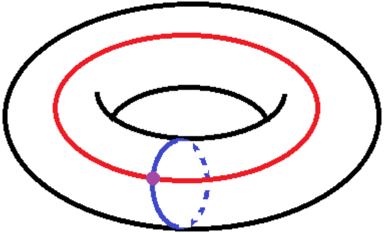
This is the smallest code which is capable of correcting arbitrary errors on a single qubit.

Table of Contents

Lattice Description of the Torus

A torus $T = S_1 \times S_1$ can be visualized using a square lattice, where the top/bottom and right/left edges are identified with each other.

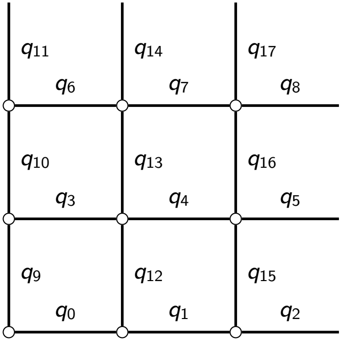




Kitaev [5] showed how it is possible to build a quantum code called a **toric code** using a *cell decomposition* of a torus.

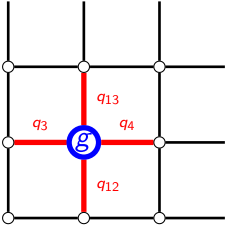
Building a Quantum Code on the Torus Lattice

Using the $m \times m$ torus lattice, we may construct a stabilizer code with $n = 2m^2$ qubits, where each qubit is identified with an **edge**.

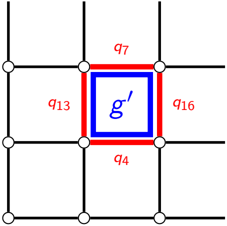


Stabilizers on the Torus Lattice

X-stabilizer generators are defined by **nodes**.
Z-stabilizer generators are defined by **plaquettes**.



$g = X_3X_4X_{12}X_{13}$



$g' = Z_4Z_7Z_{13}Z_{16}$

X- and Z-type stabilizer generators overlap on zero or two qubits, and hence will commute (a requirement for stabilizer codes).

Check Matrices for the 3×3 Toric Code

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

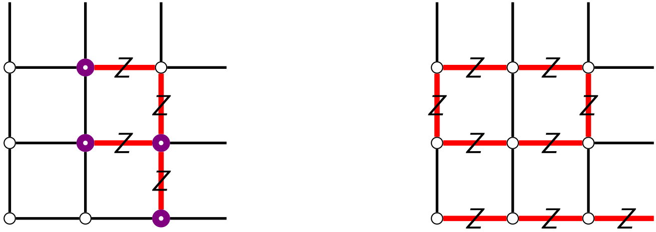
$$H_Z = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0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

Notice that H_X and H_Z for the 3×3 toric code contain the block structure of several copies of the 3-bit repetition code!

Errors and Syndromes on the Torus Lattice

A Z-error on some combination of qubit edges is detected by the X-check nodes adjacent to an *odd number* of erroneous qubits.

The H_X-**syndrome** for this Z-**error** is visualized by these nodes.



A **logical error** has undetectable syndrome; these correspond to *closed loops* on the lattice and may be trivial or nontrivial.

- A *contractible* loop is equivalent to a product of stabilizers.
- A loop that *wraps around* the lattice is a non-trivial error.

Generalizations of the Toric Code

More generally, any cell decomposition for any surface can be used to construct a quantum code in this way; these are **surface codes**.

- The topology of a surface code can be exploited for decoding.
- The lattice structure is useful for physical implementation.

The toric code is an example of a **hypergraph product** (HGP) code constructed using two repetition codes.

- A HGP code is constructed from any two classical codes $C_1 = \ker(H_1)$, $C_2 = \ker(H_2)$ of sizes $H_i = [r_i \times n_i]$.
- $H_X = [H_1 \otimes I_{n_2} | I_{r_1} \otimes H_2^T]$ and $H_Z = [I_{n_1} \otimes H_2 | H_1^T \otimes I_{r_2}]$.

References

[1] A Robert Calderbank and Peter W Shor. Good quantum error-correcting codes exist. Physical Review A, 54(2):1098, 1996.

[2] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum error correction for beginners. Reports on Progress in Physics, 76(7):076001, 2013.

[3] Daniel Gottesman. Stabilizer codes and quantum error correction. California Institute of Technology, 1997.

[4] DC Hankerson, Gary Hoffman, Douglas A Leonard, Charles C Lindner, Kevin T Phelps, Chris A Rodger, and James R Wall. Coding theory and cryptography: the essentials. CRC Press, 2000.

[5] A Yu Kitaev. Fault-tolerant quantum computation by anyons. Annals of Physics, 303(1):2–30, 2003.

[6] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information. American Association of Physics Teachers, 2002.

[7] Peter W Shor. Scheme for reducing decoherence in quantum computer memory. Physical review A, 52(4):R2493, 1995.

[8] Andrew Steane. Multiple-particle interference and quantum error correction. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 452(1954):2551–2577, 1996.

[9] Andrew M Steane. A tutorial on quantum error correction. Quantum Computers, Algorithms and Chaos, pages 1–32, 2006.