Quantum LDPC codes
Lecture 1

Nicolas Delfosse

Microsoft

PCMI Summer School 2023

Overview

- Classical codes
- Stabilizer codes
- Good stabilizer exist
- Examples of quantum LDPC codes

Classical linear codes

Error correction with repetition codes

$$001 \xrightarrow{\text{errors}} 011$$

$$001 \xrightarrow{\text{encoding}} (000)(000)(111) \xrightarrow{\text{errors}} (001)(000)(011) \xrightarrow{\text{decoding}} 001$$

$$001 \xrightarrow{\text{encoding}} (00000)(00000)(11111) \xrightarrow{\text{errors}} (01100)(00001)(01100) \xrightarrow{\text{decoding}} 000$$

Monte-Carlo simulation of repetition codes

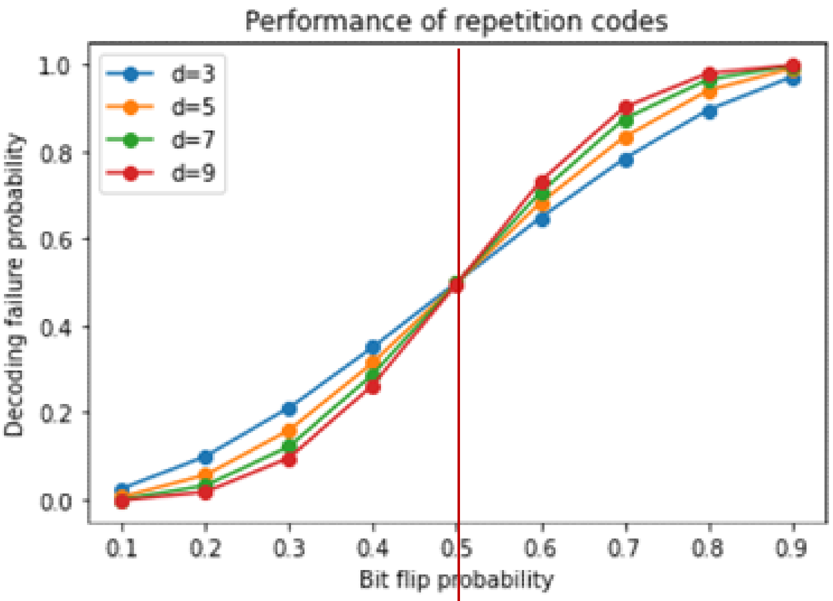
$$x = 0 \text{ or } 1 \xrightarrow{\text{encoding}} (\overbrace{xx \dots x}^{d \text{ repetitions}})$$

d	# bit-flip corrected
3	1
5	2
7	3

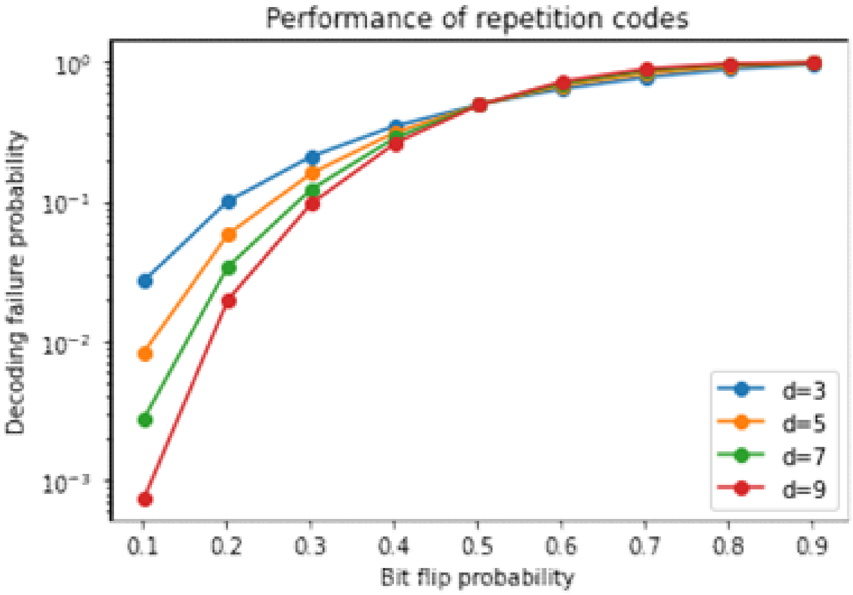
Monte-Carlo simulation:

```
Initialize N = 0.
Repeat 1,000,000 times:
    Start with the encoded bit-string (00 … 0).
    Flip each bit independently with probability p.
    Apply a majority vote decoder.
    If the decoded bit is 1, increment N.
Return N / 1,000,000
```

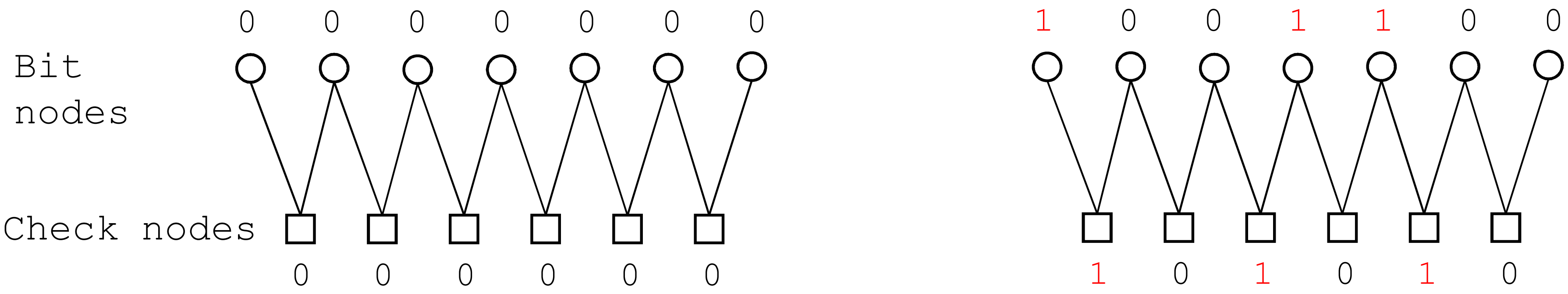
Monte-Carlo simulation of repetition codes



Code threshold = 0.5



Tanner graphs of repetition codes

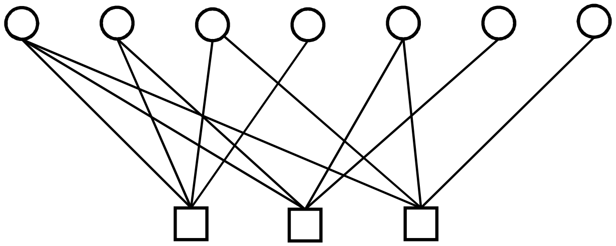


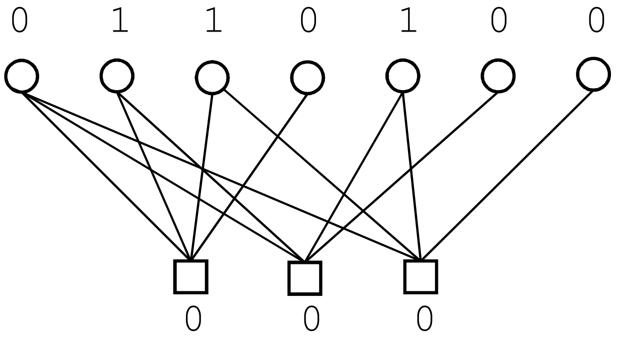
Issue: The repetition code encodes only 1 logical bit.
Solution: Use more general graphs.

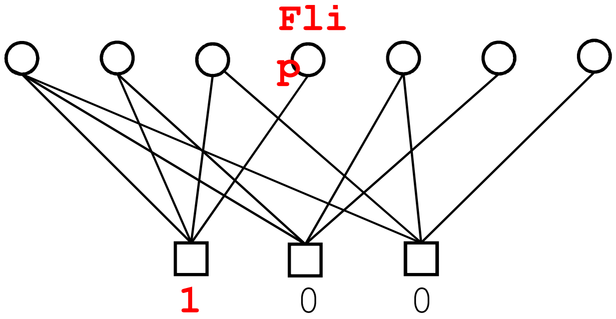
Hamming code

Solution: Using general graphs, we can simultaneously:

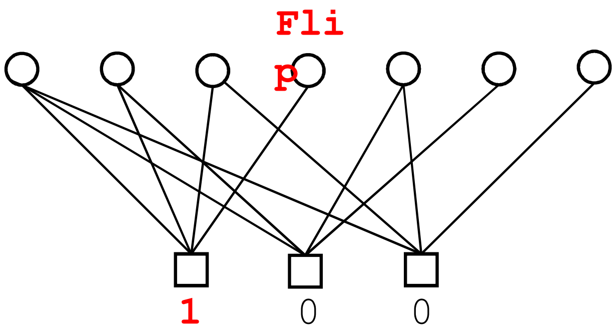
- encode many logical bits
- correct many bit-flips







Decoding Hamming code



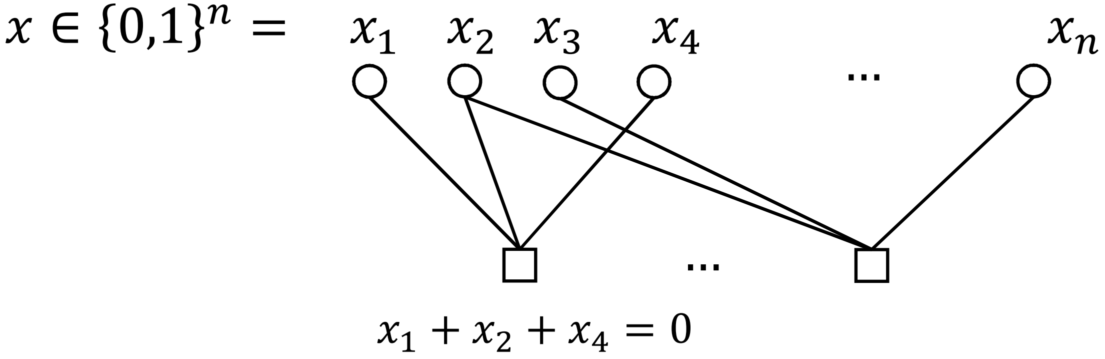
Lookup table decoder:

Checks	Correction
100	Flip 4
010	Flip 6
110	Flip 2
001	Flip 7
101	Flip 3
011	Flip 5
111	Flip 1

Issue: There is no known polynomial time decoding algorithm for general graphs
Solution: Use sparse graph

Low Density Parity-Check (LDPC) codes

Solution: LDPC codes = codes defined by low degree checks



With sparse graph, we get an efficient decoder.
Basic idea: if $x_1 + x_2 + x_4 = 1$ there is only 3 possible bit-flips
=> local decoding.

Quantum generalization

Requirements:

- Encode many logical qubits,
- Correct many Pauli errors,
- Efficient decoder,
- Fault-tolerant: Can be implemented with very noisy quantum hardware,

=> Quantum LDPC codes

Quantum stabilizer codes

Pauli operators

Ex.

- $(-i)X \otimes I \otimes Y \otimes X$
- ZZZ
- $-X_1X_3X_5X_7$

Notation.

- $\mathcal{P}_n$: Set of all n-qubit Pauli operators.

Commutation

Prop. Two Pauli operators $P, Q \in \mathcal{P}_n$ either commute or anti-commute.

Ex.

- $XIIZ$ and $ZIIZ$?
- XYZ and ZXY?
- $XIXIXIXIIIIXXXXX$ and $XIXXIIXIIIIXXIIX$?
- $XXXX$ and $YYYY$?

Notation.

$$[P, Q] = \begin{cases} 0 & \text{if they commute} \\ 1 & \text{if they anticommute} \end{cases}$$

Stabilizer codes

Def. A stabilizer code is defined to be a subspace $Q(S)$ of $(\mathbb{C}^2)^{\otimes n}$ fixed by a set S of Pauli operators.

Prop. The set S is a commutative subgroup of P_n that does not contain $-I$. Conversely, we can define a stabilizer code for each commutative subgroup of P_n that does not contain $-I$.

Remark:

- A commutative subgroup of P_n that does not contain $-I$ is called a stabilizer group.
- The elements of S are called stabilizers.
- If $S = \langle S_1, \dots, S_r \rangle$, its generators S_i are called stabilizer generators.

Exercise. Prove the proposition.

Stabilizer matrix

Ex. The five-qubit code is defined by the stabilizer matrix:

$$H = \begin{bmatrix} X & Z & Z & X & I \\ I & X & Z & Z & X \\ X & I & X & Z & Z \\ Z & X & I & X & Z \end{bmatrix}$$

Exercise. Check that the rows of this matrix generate a stabilizer group.

Number of logical qubits

Def. A set of Pauli operators $\{S_1, \dots, S_r\}$ is independent if the only product of these operators that is equal to I is the trivial product.

Prop. If $S = \langle S_1, \dots, S_r \rangle$ is generated by r independent generators, then $Q(S)$ encodes $k \coloneqq n - r$ logical qubits, i.e $\dim Q(S) = 2^{n-r}$.

Proof. Problem session.

Remark.

- n is called the code length.
- The code parameters are denoted $[[n, k]]$.

Example

Exercise. Count the number of logical qubits for the following stabilizer codes:

- $H = \begin{bmatrix} X & Z & Z & X & I \\ I & X & Z & Z & X \\ X & I & X & Z & Z \\ Z & X & I & X & Z \\ Z & Z & X & I & X \end{bmatrix}$

=> [[5, 1]]

- $H = \begin{bmatrix} X & I & X & I & X & I & X \\ I & X & X & I & I & X & X \\ I & I & I & X & X & X & X \\ Z & I & Z & I & Z & I & Z \\ I & Z & Z & I & I & Z & Z \\ I & I & I & Z & Z & Z & Z \end{bmatrix}$

=> [[7, 1]]

A code with only X type or Z type rows is called a CSS code.

Measurement of the syndrome of an error

Def. Given $S_1, \dots, S_r$, the syndrome of a Pauli error $E \in \mathcal{P}_n$ is the vector $\sigma(E) = (\sigma_1, .., \sigma_r) \in \{0,1\}^r$ such that $\sigma_i = [E, S_i]$.

Prop. Consider a system in the state $E|\psi\rangle$ where $|\psi\rangle \in Q(S)$ and $E \in \mathcal{P}_n$.

- The outcome of the measurement of S_i is $(-1)^{\sigma_i(E)}$ with probability 1.
- The state of the system after measurement is $E|\psi\rangle$.

Proof. Exercise.

Hint. What is the projector onto the $(-1)^a$-eigenspace of S_i?

Example

Exercise. Consider the stabilizer code:

$$H = \begin{bmatrix} X & I & X & I & X & I & X \\ I & X & X & I & I & X & X \\ I & I & I & X & X & X & X \\ Z & I & Z & I & Z & I & Z \\ I & Z & Z & I & I & Z & Z \\ I & I & I & Z & Z & Z & Z \end{bmatrix}$$

- What is the syndrome of $X_1X_3X_5X_7$? X_3? X_1X_2?
- $\sigma(X_1X_3X_5X_7) = (0,0,0,0,0,0)$
- $\sigma(X_3) \quad = (0,0,0,1,1,0)$
- $\sigma(X_1X_2) \quad = (0,0,0,1,1,0)$
- Can you find an errors with trivial syndrome? Is it a stabilizer?

Logical basis

Def. A logical error for a stabilizer code is a Pauli error with trivial syndrome.

It is a non-trivial logical error if it is a logical error and not a stabilizer.

Prop. For all $[[n,k]]$ stabilizer code, there exists a set of logical error of the form

$$\overline{X_1}, \overline{Z_1}, \dots, \overline{X_k}, \overline{Z_k}$$

such that

- $[\overline{X}_i,\ \overline{Z}_j] = \delta_{i,j}$
- $[\overline{X}_i,\ \overline{X}_j] = [\overline{Z}_i,\ \overline{Z}_j] = 0$

Proof. Exercise.

Hint. Apply on Gram-Schmidt process.

Example

- $H = \begin{bmatrix} X & Z & Z & X & I \\ I & X & Z & Z & X \\ X & I & X & Z & Z \\ Z & X & I & X & Z \end{bmatrix}$

Find a logical basis?

- $\overline{X_1} = XXXXX$
- $\overline{Z_1} = ZZZZZ$

Minimum distance

Def. The minimum distance of a stabilizer code, denoted d, is the minimum weight of a non-trivial logical error, i.e.

$$d = \min\{|E| \text{ such that } E \in \mathcal{P}_n \backslash S, \sigma(E) = 0\}$$

Example

Exercise. Compute the parameters of the following codes.

$$H = \begin{bmatrix} X & I & X & I & X & I & X \\ I & X & X & I & I & X & X \\ I & I & I & X & X & X & X \\ Z & I & Z & I & Z & I & Z \\ I & Z & Z & I & I & Z & Z \\ I & I & I & Z & Z & Z & Z \end{bmatrix}$$

⇒ [[7, 1, 3]]

Remark.

- When d is known, the code parameters and are denoted [[n, k, d]].

Decoder

Def. A decoder is a map $D\colon \{0,1\}^r \to \mathcal{P}_n$. We say that the decoder corrects a Pauli error E if the $D(\sigma(E)) = E \bmod S$.

Def. A minimum weight (MW) decoder is a map $D\colon \{0,1\}^r \to \mathcal{P}_n$ such that for all $\sigma \in \{0,1\}^r$, $D(\sigma)$ is a minimum weight error with syndrome σ.

Prop. A MW decoder corrects all Pauli errors E with weight $|E| \le \frac{d-1}{2}$.

Proof.

Assume that an error E occurs with $|E| \le \frac{d-1}{2}$.

The decoder returns a correction E' with syndrome $\sigma(E') = \sigma(E)$.

$\sigma(EE') = \sigma(E) + \sigma(E') = 0$ and $|EE'| \le |E| + |E'| \le d - 1$.

Therefore, $EE' \in S$ and the decoder correct E.

Example

How many errors can we correct with the following code?

$$H = \begin{bmatrix} X & I & X & I & X & I & X \\ I & X & X & I & I & X & X \\ I & I & I & X & X & X & X \\ Z & I & Z & I & Z & I & Z \\ I & Z & Z & I & I & Z & Z \\ I & I & I & Z & Z & Z & Z \end{bmatrix}$$

Good stabilizer codes

Theorem. There exists a sequence of $[[n, k, d]]$ stabilizer codes with $n \to +\infty$ and k, d linear in n. More precisely, for all $\delta \in [0,1]$ and for all $\varepsilon > 0$, we can achieve

- $\frac{k}{n} = 1 - h(\delta) - \delta \log_2 3 - \varepsilon$
- $\frac{d}{n} \geq \delta$

Proof idea. Consider the random variable

$Y_{\delta n}(C) \coloneqq$ # of Pauli errors $E \in \mathcal{P}_n \backslash S$ with $\sigma(E) = 0$ with weight $|E| \leq \delta n$

1. Show that if $\frac{k}{n}$ is small enough then $\mathbb{E}(Y_{\delta n}) \to 0$ when $n \to +\infty$.
2. This proves the existence of a family of codes with $Y_{\delta n}(C) = 0$.
3. By definition $Y_{\delta n}$, this shows that $d > \delta n$.

Lemmas – Counting stabilizer codes

A stabilizer group for a $[[n,k]]$ stabilizer code is of the form $S = \langle S_1, \dots, S_{n-k} \rangle$.

Lemma. The number of $[[n,k]]$ stabilizer code is

$$2^{n-k} \prod_{i=0,\dots,n-k-1} \frac{\left(2^{2n-i} - 2^i\right)}{\left(2^{n-k} - 2^i\right)}$$

Proof. # stabilizer codes =

$$\frac{\text{\# choices for } n\text{–}k \text{ independent stabilizer generators}}{\text{\# generating sets of a fixed stabilizer group}}$$

- 2^{n-k} = # choices for the phase ± 1 of each S_i
- # choices for the 1st generator = $2^{2n} - 1$.
- # choices for the 2nd generator = $2^{2n-1} - 2$.
- # choices for the 2nd generator = $2^{2n-2} - 4$.
- …
- => # choices for *n*–*k* independent stabilizer generators = $2^{n-k} \prod_{i=0,\dots,n-k-1} \left(2^{2n-i} - 2^i\right)$
- # generating sets of $\langle S_1, \dots, S_{n-k} \rangle$ = $\prod_{i=0,\dots,n-k-1} \left(2^{n-k} - 2^i\right)$

Lemmas – Counting stabilizer codes

Lemma. The number of $[[n,k]]$ stabilizer code is

$$2^{n-k}\prod_{i=0,\ldots,n-k-1}\frac{\left(2^{2n-i}-2^{i}\right)}{\left(2^{n-k}-2^{i}\right)}$$

Lemma. Let $E \neq I$. The number of $[[n,k]]$ stabilizer group such that $\sigma(E)=0$ is

$$2^{n-k}\prod_{i=0,\ldots,n-k-1}\frac{\left(2^{2n-i-1}-2^{i}\right)}{\left(2^{n-k}-2^{i}\right)}$$

Proof. Similar.

Good stabilizer codes - Proof

Proof. We can write $Y_{\delta n}(Q)$ as

$$Y_{\delta n}(Q) = \sum_{\substack{E \in \mathcal{P}_n \setminus S \\ |E| \leq \delta n}} X_E(Q)$$

where

$$X_E(Q) = \begin{cases} 0 \text{ if } \sigma(E) \neq 0 \\ 1 \text{ if } \sigma(E) = 0 \end{cases}$$

By linearity of the expectation, we have

$$\mathbb{E}(Y_{\delta n}) = \sum_{\substack{E \in \mathcal{P}_n \setminus S \\ |E| \leq \delta n}} \mathbb{E}\big(X_E(Q)\big)$$

Moreover, $\mathbb{E}\big(X_E(Q)\big) = \mathbb{P}(\sigma(E) = 0)$.

Good stabilizer codes - Proof

Lemma. For all $E \neq I$, we have$\mathbb{P}(\sigma(E) = 0) \leq 2^{-(n-k)}$.

Proof.

$$\mathbb{P}(\sigma(E) = 0) = \frac{\text{number of } [[n,k]] \text{ codes with } \sigma(E) = 0}{\text{number of } [[n,k]] \text{ codes}} = \prod_{i=0,\ldots,n-k-1} \frac{\left(2^{2n-i-} \quad - 2^i\right)}{\left(2^{2n-i} - 2^i\right)} \leq 2^{-(n-k)}$$

Application.

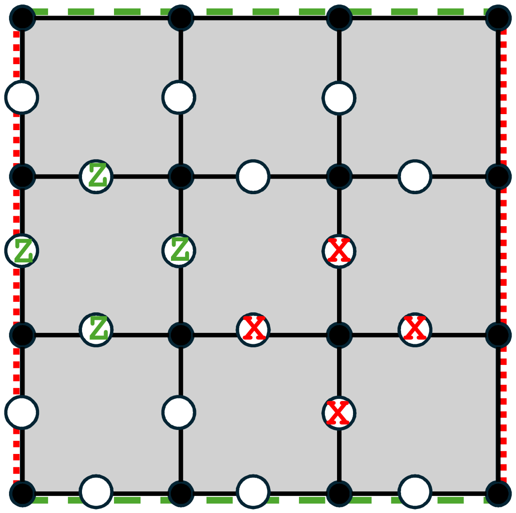
$$\mathbb{E}(Y_{\delta n}) = \sum_{\substack{E \in \mathcal{P}_n \backslash \mathrm{S} \\ |E| \leq \delta n}} \mathbb{P}(\sigma(E) = 0) \leq 2^{-(n-k)} \sum_{i \leq \delta n} 3^i \binom{n}{i} \leq \text{poly}(n) \cdot 2^{-(n-k)+n \quad (\delta)+n\delta \log_2 3}$$

$$= \text{poly}(n) \cdot 2^{-n\left((1-h(\delta)-\delta \,\text{lo} \quad _2 3)-\frac{k}{n}\right)}$$

which goes to 0 if $\frac{k}{n} = 1 - h(\delta) - \delta \log_2 3 - \varepsilon$ with $\varepsilon > 0$. This concludes the proof.

Example of LDPC codes

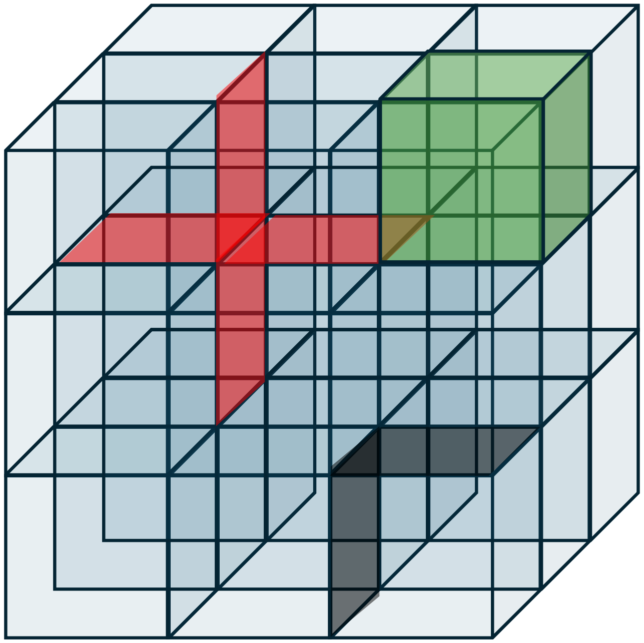
Example – Kitaev's toric code



Consider a cellulation of the torus.

- Place a qubit on each edge.
- Define a X generator for each vertex.
- Define a Z generator for each face.

Example - 3D toric code



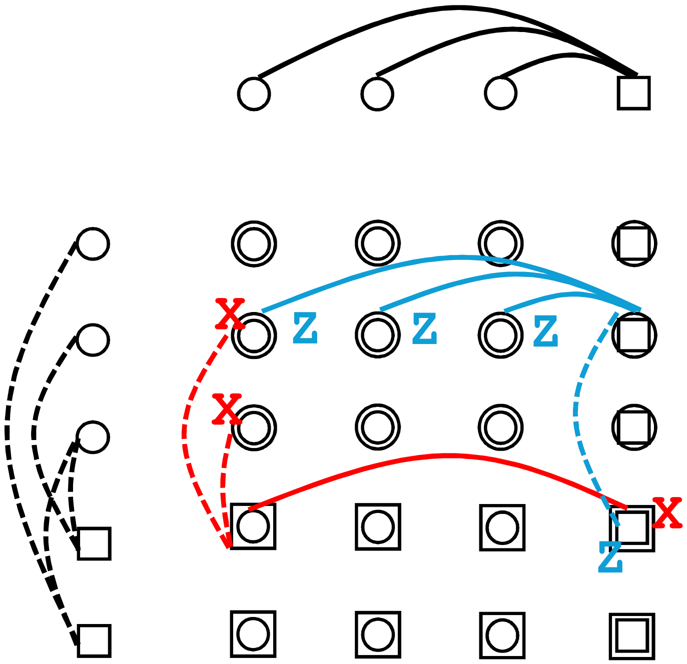
Consider a cellulation of a 3-dim manifold.

- Place a qubit on each face.
- Define a X generator for each edge.
- Define a Z generator for each 3-cell.

Or

- Place a qubit on each edge.
- Define a X generator for each vertex.
- Define a Z generator for each

Example – Hypergraph product code



Consider two bipartite graph

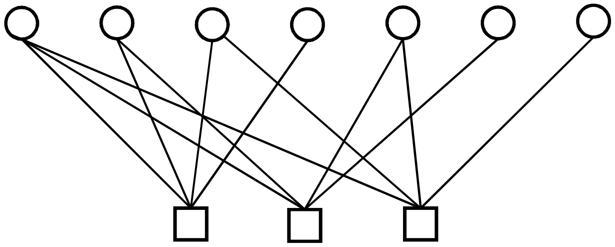
- Place a qubit on each circle-circle.
- Place a qubit on each square-square.
- Define a X generator for each square-circle.
- Define a Z generator for each circle-square.

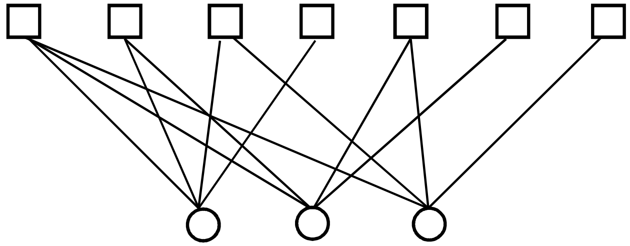
Comparison of the parameters

Code	k	d
2D toric codes	constant	$\propto \sqrt{n}$
3D toric codes	constant	$\propto n^{1/3}$
HGP codes	$\propto n$	$\propto \sqrt{n}$

Hypergraph Product (HGP) Codes

Linear code and transposed code





Linear code parameters
$[n, k, d]$

- n = # bits,
- r = # checks
- $k = \dim C$
- $d = \min\{|x|,\ x \in C, x \neq 0\}$

is the minimum distance.

Transposed code with parameters
$[n^T, k^T, d^T]$

- $n^T = r$
- $r^T = n$
- $k^T = k + n - r$

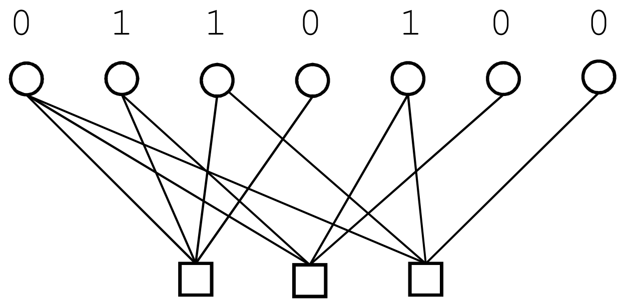
Product of two linear codes $C_1 \otimes C_2$

Def. A codeword of $C_1 \otimes C_2$ is bitstring forming a $n_1 \times n_2$ matrix x such that

- each row of x is in C_1,
- each column of x is in C_2,

Ex.

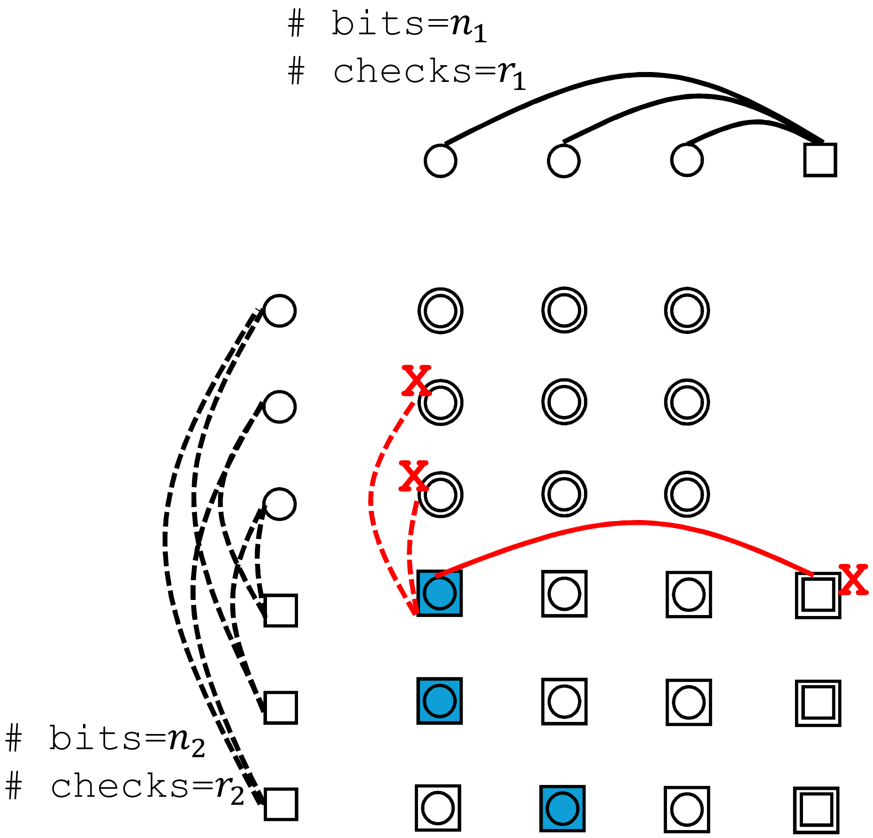
$$\begin{matrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{matrix}$$



is in the product of the Hamming code and the 3-repetition code.

Prop. The dimension of $C_1 \otimes C_2$ is $k_1 k_2$.

Number of independent generators



Lemma. The number of independent X generators is

$$n_1 r_2 - k_1 k_2^T$$

where k_2^T the dimension of the code obtained by swapping bits and checks.

Proof. The term $n_1 r_2$ is the total number of X checks.

If a product of X generators (blue) is equal to I, then:

- each circle-circle qubit is a vertical check for this product
- each square-square qubit is a horizontal check for this product.

Therefore, trivial products of X generators correspond to codewords of the classical product code $C_1 \otimes C_2^T$.

This proves that there are $k_1 k_2^T$ independent relations between the X

Number of logical qubits of HGP codes

Lemma. The number of independent X generators is

$$n_1 r_2 - k_1 k_2^T$$

Lemma. The number of independent Z generators is

$$r_1 n_2 - k_1^T k_2$$

Theorem. For HGP codes, we have

- $n = n_1 n_2 + r_1 r_2$
- $k = k_1 k_2 + k_1^T k_2^T$
- $d \geq \min(d_1, d_2, d_1^T d_2^T)$