Quantum error correction

- Explicit construction of error correcting superoperator
- Distance of QECC, Bounds on QECC (Qu. Hamming bound & Qu. Singleton bound) w/o proof
- Classical linear codes
- CSS codes
  - Steane code
- Stabilizer codes
  - General formulation & symplectic notation
  - Shor code, Steane code, CSS codes
- Subsystem codes
- Other codes: LDPC codes, color codes
- Threshold theorem (without proof)
- Universal fault-tolerant quantum computation
- Magic states for T-gates
- Magic state distillation

learned that necessary condition on codespace $\{|\bar{i}\rangle\}$ & error operators $\{M_\mu\}$ is that

$$\langle \bar{i}| M_\mu^\dagger M_\nu |\bar{j}\rangle = C_{\mu\nu}\,\delta_{ij} \qquad \forall i,j,\mu,\nu$$

↑ hermitian matrix that is independent of i,j.

To show that this condition is also sufficient, let us explicitly construct the recovery superoperator that reverses the error.

Since $C_{\mu\nu}$ is a hermitian matrix, we can choose environmental basis $\{|\mu\rangle_E\}$ where $C_{\mu\nu} = C_\mu \delta_{\mu\nu}$ is diagonal.

$$\Rightarrow \langle \bar{j}| M_\delta^\dagger M_\mu |\bar{i}\rangle = C_\mu \delta_{\delta\mu}\delta_{ij},$$

where $\sum_\mu C_\mu = 1$ from $\sum_\mu M_\mu^\dagger M_\mu = \mathbb{I}$.

For each ν with $C_\nu \neq 0$, let's define the Kraus operator

$$R_\nu = \frac{1}{\sqrt{C_\nu}} \sum_i |\bar{i}\rangle\langle \bar{i}| M_\nu^\dagger .$$

Then, R_ν acts on state $M_\mu |\bar{i}\rangle$ as follows

$$R_\nu M_\mu |\bar{i}\rangle = \frac{1}{\sqrt{C_\nu}} \sum_j |\bar{j}\rangle \underbrace{\langle \bar{j}| M_\nu^\dagger M_\mu |\bar{i}\rangle}_{= C_\nu \delta_{\nu\mu} \delta_{ij}} =$$

$$= \sqrt{C_\nu}\, \delta_{\mu\nu} |\bar{i}\rangle .$$

The superoperator defined by $\{R_\nu\}$ as $S \mapsto \sum_\nu R_\nu S R_\nu^\dagger$ thus disentangles the codespace from the environment E and instead entangles E with the ancilla A:

$$\sum_{\mu,\nu} R_\nu M_\mu |\bar{i}\rangle |\mu\rangle_E |\nu\rangle_A = |\bar{i}\rangle \otimes \left(\sum_\nu \sqrt{C_\nu}\, |\nu\rangle_E |\nu\rangle_A \right)$$

$$\equiv |\bar{i}\rangle \otimes |\text{stuff}\rangle_{EA} .$$

Remains to test completeness of $\{R_\nu\}$:

$$\sum_\nu R_\nu^+ R_\nu = \sum_{\nu,i} \frac{1}{C_\nu} M_\nu |\bar{i}\rangle\langle\bar{i}| M_\nu^+$$

$\sum_\nu P_\nu \equiv$ orthogonal projection onto space of states that can be reached by errors $\{M_\nu\}$ acting on codewords $\{|\bar{i}\rangle\}$.

By including one additional projector onto the complement $I - \sum_\nu P_\nu$ into the set of $\{R_\nu\}$, we ensure completeness.

NB:

- Nondegenerate codes fulfill the condition

$$\langle\bar{j}| E_b^+ E_a |\bar{i}\rangle = \delta_{ab}\,\delta_{ij}$$

i.e., $C_{ab} = \delta_{ab}$ then.

They map each Pauli error E_a onto mutually orthogonal error subspaces. Degenerate codes (like Shor's code)

feature the same error syndrome for different errors.

The recovery Kraus operator for nondegenerate code is thus

$$R_a = \sum_i |\bar{i}\rangle\langle\bar{i}| E_a^\dagger$$

for each E_a in set of correctable errors $\mathcal{E} = \{E_a\}$.

- May view error correction procedure as two-step process:
  - collective measurement to determine error syndrome
  - based on measurement outcome, apply unitary U_f that reverses the error (for nondeg. codes $\hat{=}$ application of $E_a^\dagger$)

- But, measurement is <u>not needed</u>, but ancillas are <u>needed</u>. Ancillas serve as depository for entropy that is inserted into code block by error. It „heats" as the data „cools". Need continuous supply of ancillas that are discarded (or measured) after use.

Distance & bounds of QECC

Distance d = minimum weight of a Pauli operator E_a such that $\langle \bar{i} | E_a | \bar{j} \rangle \neq C_a \delta_{ij}$

$\hat{=}$ minimum weight of Pauli operator that induces logical transitions.

Since QECC condition is $\langle \bar{i} | E_a^\dagger E_b | \bar{j} \rangle = C_{ab} \delta_{ij}$, it follows that a code of distance d can correct t errors when $d = 2t+1$. Reason: if E_a, E_b have weight at most t, then $E_a^\dagger E_b$ has at most weight $2t$ and if $d > 2t$ (i.e. $d \geq 2t+1$) then the distance criterion implies that the QECC condition holds.

Bounds on code parameters $[[n, k, d]]$:

We want a high rate $R = \frac{k}{n}$ and large distance d.

There exist some bounds of what one can optimally achieve.

(1) Quantum Hamming bound for nondegenerate codes

$[[n, k, d]]$ with distinct syndromes for every error.

- 3 possible errors per qubit n: X_j, Y_j, Z_j
- In each block of n qubits there are $\binom{n}{j}$ distinct ways of choosing which qubit suffers from an error.

$\Rightarrow$ total number of possible errors up to weight t

$$N(t) = \sum_{j=0}^{t} 3^j \binom{n}{j}.$$

For k encoded qubits there are 2^k linearly independent codewords.

error operators up to weight t

For nondegenerate codes all $E_a|\bar{i}\rangle$ are linearly independent (every codeword gets mapped onto a different, linearly independent set of states by errors). There needs to be enough space in Hilbert space of n qubits, which has size 2^n, and thus

$$2^n \geq 2^k \cdot \sum_{j=0}^{t} 3^j \binom{n}{j} = 2^k N(t)$$

$$\Leftrightarrow \boxed{2^{n-k} \geq N(t) \equiv \sum_{j=0}^{t} 3^j \binom{n}{j}.}$$

Quantum Hamming bound for nondegenerate codes

Example:

- $k = t = 1$: $2^{n-1} \geq 1 + 3n$

is satisfied for $n \geq 5$. $n = 5$ satisfies bound: $1 + 15 = 16$. A nondegenerate code $[[5, 1, 3]]$ would thus be perfect (no wasted space in qubit Hilbert space).

Another useful bound is the Quantum Singleton bound:

$$n - k \geq 2(d-1)$$

Quantum version of (classical) Singleton bound $n-k \geq d-1$.

$\Rightarrow$ cannot increase rate $\frac{k}{n}$ arbitrarily w/o reducing distance.

One calls a $[[n, k, d]]$ code „good" if both the rate $R = \frac{k}{n}$ and the „error probability" $p = \frac{t}{n}$,

where $t = \frac{d-1}{2}$, approach a nonzero limit as $n \to \infty$.

Turns out that good quantum codes exist & they can be chose to be nondegenerate.

Classical linear codes

In a binary linear code C the 2^k codewords $y \in C$ form a k-dimensional closed linear subspace C of the binary vector space $\mathbb{F}_2^n \supseteq C$.

The space C is spanned by a basis of k vectors $\{v_1, v_2, \ldots, v_k\}$. An arbitrary codeword is given by

$$v(\underbrace{\alpha_1, \ldots, \alpha_k}_{\in \{0,1\}^{\otimes k}}) = \sum_{i=1}^{k} \alpha_i v_i \in C \subseteq \mathbb{F}_2^n .$$

Example: 3-repetition code: $n = 3$, $k = 1$ $\Rightarrow$ 2 codewords

C is spanned by vector $v_1 = (1,1,1)$.

Codewords $v(0) = (0,0,0)$

$v(1) = (1,1,1)$

The length-n row vector $v(\alpha_1, \ldots, \alpha_k)$ encodes the k-bit message $\alpha = (\alpha_1, \ldots, \alpha_k)$.

Generator matrix G (note that Preskill's def. used here is related to NRC as $G = G_{NRC}^T$).

The k-basis vectors $v_1, \dots, v_k$ can be collected into the $k \times n$ generator matrix

$$G = \begin{pmatrix} — & v_1 & — \\ — & v_2 & — \\ & \vdots & \\ — & v_k & — \end{pmatrix} \quad \begin{matrix} \} k \\ \\ \\ \\ \\ \} n \end{matrix} .$$

Then,

$$\underbrace{v^T(\alpha)}_{\text{n-dim. row vector that encodes the k-bit message $(\alpha_1, \dots, \alpha_k)^T$.}} = \alpha G = (\alpha_1, \dots, \alpha_k) \begin{pmatrix} — & v_1 & — \\ & \vdots & \\ — & v_k & — \end{pmatrix}$$

Alternatively, we can specify the k-dimensional subspace C as the kernel of a $(n-k) \times n$ dimensional matrix H, i.e., by specifying $(n-k)$ linear constraints:

column vector of encoded message

$$H\vec{v} = 0 \qquad \forall v \in C$$

H is a $(n-k)\times n$ matrix called parity check matrix.

H has $(n-k)$ linearly independent rows that span the complement of the code subspace C:

$$H = \begin{pmatrix} - & v_{n-k+1} & - \\ & \vdots & \\ - & v_n & - \end{pmatrix}$$

$$Hv = \begin{pmatrix} - & v_{n-k+1} & - \\ & \vdots & \\ - & v_n & - \end{pmatrix} \begin{pmatrix} | \\ v \\ | \end{pmatrix} = 0$$

for $v \in C$ as $v_1, \ldots, v_k$ are all $\perp$ to $v_{n-k+1}, \ldots, v_n$.

Note that two binary vectors are orthogonal if they „collide" (both take the value 1) at an even number of locations.

From the above it holds that

$$H G^T = 0$$

$$\Leftrightarrow \begin{pmatrix} - & v_{n-k+1} & - \\ & \vdots & \\ - & v_n & - \end{pmatrix} \begin{pmatrix} | & & | \\ v_1 & \cdots & v_k \\ | & & | \end{pmatrix} = 0$$

The rows of G are orthogonal to the rows of H.

Example. 3-repetition code, $n = 3$, $k = 1$

Saw $G = \underset{n}{(1\ 1\ 1)}$ $\Rightarrow$ from $HG^T = 0$,

we find $H = \underset{n}{\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}}$ $n-k=2$

Note that rows of H are $\perp$ to row of $G = (1,1,1)$ as they collide in two locations.

$(1\ 1\ 0)\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = 1+1 = 0$, $(1\ 0\ 1)\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = 1+1 = 0$ ✓

Error detection using H

Errors are nonzero n-bit strings $e \in \{0,1\}^{\oplus n} \neq 0$.

An encoded message becomes (a n-dim. column vector)

$$\tilde{y} = y + e$$

and the error syndrome is obtained from

$$H\tilde{y} = \underbrace{Hy}_{=0} + He = He$$

Error recovery requires that all errors e_i we wish to correct have different syndromes. If we have identified the error e from analyzing He, we can correct it by adding e to the corrupted message:

$$\tilde{y} = (y+e)+e = y \quad \text{as} \quad 2e = 0 \pmod 2.$$

In contrast, if $He_1 = He_2$ for $e_1 \neq e_2$, we may

erroneously apply e_2 after error e_1 occurred

$$\tilde{y} = (y + e_1) + e_2 \neq y \text{, but}$$

$$H(y + e_1 + e_2) = 0 \text{, since } He_1 = He_2 \Leftrightarrow H(e_1 - e_2) = 0$$

$$\Leftrightarrow H(e_1 + e_2) = 0 .$$

$e_2 = -e_2$ (due to mod 2 addition)

$\Rightarrow y + e_1 + e_2 \in C$ is a valid codeword that is different from C and our attempted correction has produced a logical error.

Example: 3-repetition code

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad e \in \{(1,0,0)^T, (0,1,0)^T, (0,0,1)^T\} :$$

e	$(0,0,1)$	$(0,1,0)$	$(1,0,0)$
He	$(0,1)$	$(1,0)$	$(1,1)$

← error syndrome = binary representation of error location $i = 1, 2, 3$.

Dual of a code

A k-dim. code C is defined by $k \times n$ generator matrix G and $(n-k) \times n$ dimensional parity check matrix H that obey

$$H G^T = 0 \Rightarrow G H^T = 0.$$

The dual code $C^\perp$ is a $(n-k)$-dim. code with generator matrix $G^\perp = H$ and parity check matrix $H^\perp = G$ (switching the roles of G and H).

In other words $C^\perp$ is the orthogonal complement of C in $\mathbb{F}_2^n$.

Note, vector $v \in \mathbb{F}_2^n$ is self-orthogonal if it has even weight, so C and $C^\perp$ can intersect.

Code is weakly self-dual if $C \subseteq C^\perp$ and strictly self-dual if $C = C^\perp$, which is possible if $n = 2k$.

Useful identity (when deriving CSS codes):

We will need the following identity later

$$\sum_{v \in C} (-1)^{v \cdot u} = \begin{cases} 2^k & , u \in C^\perp \\ 0 & , u \notin C^\perp \end{cases}$$

Nontrivial content is that sum vanishes for $u \notin C^\perp$.

To show this we use that

$$\sum_{v \in \{0,1\}^{\otimes k}} (-1)^{v \cdot w} = 0 \quad \text{if } w \neq 0$$

Example: $k = 2$, $v \in \{(00), (01), (10), (11)\}$

e.g. $w = (10)$

$$\Rightarrow \sum_{v \in \{0,1\}^{\otimes k}} (-1)^{v \cdot w} = 1 + 1 - 1 - 1 = 0$$

Another example: $k = 3$, $v \in \{(000), (001), (010), (011), (100), (101), (110), (111)\}$

$w = (101)$ $\qquad v \cdot w = 1$

$$\Rightarrow \sum_{v \in \{0,1\}^{\otimes 2}} (-1)^{v \cdot w} = 1 - 1 + 1 - 1 - 1 + 1 - 1 + 1 = 0$$

From $\sum_{v \in \{0,1\}^{\otimes 2}} (-1)^{v \cdot w} = 0$ if $w \neq 0$, it follows that

if we can express $v \in C$ as

$$v = \alpha G \quad \text{with } 2\text{-dim. vector}$$

$$\alpha = (\alpha_1, \ldots, \alpha_2)$$

that

$$\sum_{v \in C} (-1)^{v \cdot u} = \sum_{\alpha \in \{0,1\}^{\otimes 2}} (-1)^{\alpha \cdot G u} = 0 \quad \text{for } Gu \neq 0.$$

Since G is the generator matrix of C and the parity check matrix of $C^\perp$ (thus codewords $u \in C^\perp$ obey $Gu = 0$), it holds that

$$\sum_{v \in C} (-1)^{v \cdot u} = 0 \quad \text{for } u \notin C^\perp.$$

For $u \in C^{\perp}$ for which $Gu = 0$, it is obvious that the sum equals 2^n. □

CSS codes (Calderbank-Shor-Steane codes)

Uses concept of dual code. CSS codes are subclass of more general class of stabilizer codes.

The CSS construction uses as starting points

two classical codes C_1 and C_2, such that $C_2 \subset C_1$

and C_1 and $C_2^{\perp}$ can both correct t errors.

C_1 is $[n, k_1]$ code with $(n-k_1) \times n$-dim.

parity check matrix $H_1 = \begin{pmatrix} - & v_{n-k_1+1} & - \\ & \vdots & \\ - & v_n & - \end{pmatrix}$.

C_2 is subcode of C_1 with $(n-k_2) \times n$-dim.

parity check matrix H_2:

$$H_2 = \begin{pmatrix} — & v_{n-r_1+1} & — \\ & \vdots & \\ — & v_n & — \\ — & v_1 & — \\ & \vdots & \\ — & v_{r_1-r_2} & — \end{pmatrix}$$

first $n-r_1$ match H_1

r_1-r_2 additional constraints making $C_2 \subset C_1$.

Each codeword $y \in C_2$ is also $y \in C_1$:

$$H_2 y = 0 \Rightarrow H_1 y = 0,$$

but the $y \in C_2$ obey some additional linear constraints arising from the last $r_1 - r_2$ rows of H_2 (they are orthogonal to those row vectors as well).

Subcode C_2 defines an equivalence relation on C_1, the equivalence classes are the cosets of C_2 in C_1:

$u, v \in C_1$ are equivalent (= belong to the same coset), iff $\exists\, w \in C_2$ such that $u = v + w$:

$(=)$ $u \equiv v$ iff $u = v + w$ for a $w \in C_2$.

There are $|C_1|/|C_2| = 2^{g_1 - g_2}$ cosets.

The CSS code $CSS(C_1, C_2)$ associates a codeword with every equivalence class ($\equiv$ every coset):

$$|\bar{v}\rangle = |v + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} |v + w\rangle$$

($|\bar{v}\rangle$: $\in CSS(C_1, C_2)$; v: $\in C_1$; $\sqrt{|C_2|} = 2^{g_2}$)

for every codeword $v \in C_1$. It is clear that

- $|\bar{v}_1\rangle = |\bar{v}_2\rangle$ if $v_1 \equiv v_2$ i.e., $v_1 = v_2 + w$ for some $w \in C_2$.
- $\langle \bar{v}_1 | \bar{v}_2 \rangle = 0$ if v_1 and v_2 lie in different equivalence classes as for no $w_1, w_2 \in C_2$ we find $v_1 + w_1 = v_2 + w_2 \Rightarrow |v_1 + w_1\rangle \perp |v_2 + v_2\rangle$ are orthonormal $\forall\, w_1, w_2$.

Consider a bitwise Hadamard tf on $|\bar{v}\rangle$:

$$H^{\otimes n}: |\bar{v}\rangle_F \equiv \frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} |v + w\rangle$$

$$\downarrow$$

$$|\bar{v}\rangle_P = \frac{1}{\sqrt{2^n}} \sum_{u \in \{0,1\}^{\otimes n}} \frac{1}{\sqrt{2^{k_2}}} \sum_{w \in C_2} (-1)^{u \cdot v} (-1)^{u \cdot w} |u\rangle$$

$$= \frac{1}{\sqrt{2^{n-k_2}}} \sum_{u \in C_2^\perp} (-1)^{u \cdot v} |u\rangle$$

identity above:

$$\sum_{w \in C_2} (-1)^{w \cdot u} = \begin{cases} 2^{k_2}, & u \in C_2^\perp \\ 0, & u \notin C_2^\perp \end{cases}$$

Bitwise Hadamard tf on $|\bar{v}\rangle$ produces state that is an equal superposition of words in the dual code $C_2^\perp$, weighted by phases $(-1)^{u \cdot v}$.

Note that $|\bar{v}\rangle_P$ depends only on coset that v represents,

i.e., $|\bar{v}\rangle_p \equiv |\bar{v} + C_2\rangle_p$ as shifting by $w \in C_2$ has no effect on $(-1)^{u\cdot(v+w)} = (-1)^{u\cdot v}(-1)^{u\cdot w}$ as $u\cdot w = 0$ if $u \in C_2^\perp$ and $w \in C_2$.

Correcting bit-flip & phase flip errors

Suppose C_1 and $C_2^\perp$ can both correct t errors, or more generally C_1 can correct t_F errors & C_2 can correct t_p errors. The code distances are the

$$d_1 \geq 2t_F + 1$$

$$d_2^\perp \geq 2t_p + 1 .$$

The, $CSS(C_1, C_2)$ can correct t_F bit flips & t_p phase flips and thus has distance

$$d \geq \min\left(d_1, d_2^\perp\right).$$

Every Pauli can be expressed as product of bit flip X & phase flip Z operator.

① Correcting bit flips

Let $e \in \{0,1\}^{\otimes n}$ be some error denoting bit flip operator $E_e^F = \prod_{\{e_i=1\}} X_i^{e_i}$ $\quad (X_i^0 = I)$.

Then, $E_e^F : |v\rangle \longrightarrow |v+e\rangle$.

$$\Rightarrow \quad E_e^F : |\bar{v}\rangle_F = \frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} |v+w\rangle$$

$$\downarrow$$

$$\frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} |v+w+e\rangle$$

To diagnose bit flip errors perform unitary U_f on data & ancilla (need suff. many ancillas to store syndrome for code C_1 and initialize ancilla qubits in state $|0\rangle$):

"$(n-k_1)$ ancillas, one for each row of H_1.

$$|v\rangle \otimes |0\rangle_A \longrightarrow |v\rangle \otimes |H_1 v\rangle_A$$

(can be implemented with CNOT gates only).

$$\Leftarrow |v+w+e\rangle |0\rangle_A \longrightarrow |v+w+e\rangle |H_1 e\rangle$$

as $v+w \in C_1$ and thus $H_1(v+w) = 0$.

Measure ancillas to obtain bit flip syndrome $H_1 e$ and flip bad bits by application of $\prod_i X_i^{e_i}$.

② Correcting phase flips

Phase flip error : $e \in \{0,1\}^{\otimes n}$ defining phase flip operator $E_e^P = \prod_i Z_i^{e_i}$ $(Z^0 = I)$.

Then,

$$E_e^P : |v\rangle \longrightarrow (-1)^{v \cdot e} |v\rangle .$$

In the Hadamard rotated basis, this becomes

$$E_e^P : |u\rangle \longrightarrow |u+e\rangle .$$

Thus, to count phase errors we first perform a Hadamard tf to rotate from F to P basis:

$$E_e^P : |\bar{v}\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{v \in C_2} |v + w\rangle$$

$$\downarrow$$

$$\frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} (-1)^{(v+w)\cdot e} |v + w\rangle$$

$$\downarrow$$ Now, perform bitwise Hadamard tf

$$\frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{2^{k_2}}} \sum_{u \in \{0,1\}^{\otimes n}} \sum_{w \in C_2} (-1)^{(v+w)(e+u)} \overbrace{|u\rangle}^{=u'} \overbrace{}^{=u'+e} =$$

After H tf the state is a superposition of codewords in P basis (up to errors).

$$= \frac{1}{\sqrt{2^{n-k_2}}} \sum_{u' \in C_2^\perp} (-1)^{v \cdot u'} |u' + e\rangle$$

↖ phase flip acts as bit flip in H tf. state.

identity $\sum_{w \in C_2} (-1)^{w \cdot u'} = \begin{cases} 2^{k_2}, & u' \in C_2 \\ 0, & u' \notin C_2^\perp \end{cases}$

Can correct for phase flips by measuring syndrome of $C_2^\perp$ via its parity check matrix $H_2^\perp = G_2$ using ancilla:

$$|v\rangle|0\rangle_A \longrightarrow |v\rangle|G_2 v\rangle_A$$

Generator matrix G_2 of C_2 is parity check matrix $H_2^\perp$ of $C_2^\perp$.

Correct state by application $\prod_i X_i^{e_i}$ to state and using Hadamard tf.

$\downarrow$ error correction $\prod_i X_i^{e_i} |u'+e\rangle \to |u'\rangle \quad \forall u'$

$$\frac{1}{\sqrt{2^{n-g_2}}} \sum_{u' \in C_2^\perp} (-1)^{v \cdot u'} |u'\rangle$$

$\downarrow$ Hadamard tf ($H^2 = I$).

$$\frac{1}{\sqrt{2^{g_2}}} \sum_{w \in C_2} |v + w\rangle$$

Summary:

Separate syndromes to measure bit flip & phase flip errors: $C_2 \subset C_1$, $C_2^\perp$, C_1 correct t errors:

$|\bar{v}\rangle = \frac{1}{|C_2|} \sum_{w \in C_2} |v+w\rangle$ are codewords

$2^{n_1 - n_2}$ dim. code = 1 codeword for each coset of C_2 in C_1.

Example: 7-qubit Steane code

$$[[n, k, d]] = [[7, 1, 3]].$$

Constructed from classical 7-bit Hamming code $[7, 4, 3]$:

with 3×7 dim. parity check matrix

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

Distance $d = 3$:

- weight-3 string (1110000) passes parity check (first 3 columns are linearly dependent).
- $He_1 \neq 0$ with e_1 being weight 1 as all columns are nontrivial.
- $H(e_1 + e_1') \neq 0$ with e_1, e_1' being weight-1 errors as all columns are distinct (linearly independent).

- Note that the rows of H pass parity check and are also in the code (are orthogonal to themselves).

Generator matrix of $[7,4,3]$ code reads

$$G = \left(\begin{array}{ccccccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array} \right) \equiv H$$

Obeys $H G^T = 0$.

First three rows coincide with H (as rows of H pass parity check). Last row is the shift-3 codeword (1110000).

The dual of the Hamming code is the $[7,3,4]$ code generated by H. It holds that $C^\perp \subset C$. $C^\perp$ is even subcode of Hamming code with codewords

basis even weight.

⇒ odd codeword (1110000) is representative of nontrivial coset of the even subcode.

CSS construction:

$$C_1 = C$$

$$C_2 = C^\perp \subset C$$

↳ even subcode of C

$$\Rightarrow C_2^\perp = C_1$$

⇒ use Hamming parity checks to detect bit flips in $\mathcal{F}$ basis & phase flips in $\mathcal{P}$ basis (after H tf.)

$\mathcal{F}$ basis:

$$|\bar{0}\rangle_{\mathcal{F}} = \frac{1}{\sqrt{8}} \sum_{\substack{\text{even } v \\ \in \text{Hamm}}} |v\rangle$$

$$|\bar{1}\rangle_F = \frac{1}{\sqrt{8}} \sum_{\substack{\text{odd } v \\ \in \text{Hamm}}} |v\rangle$$

$$H^{\otimes 7} : |\bar{0}\rangle_F \to |\bar{0}\rangle_P \equiv \frac{1}{4} \sum_{v \in \text{Hamm}} |v\rangle = \frac{1}{\sqrt{2}} \left(|\bar{0}\rangle_F + |\bar{1}\rangle_F \right)$$

$$|\bar{1}\rangle_F \to |\bar{1}\rangle_P \equiv \frac{1}{4} \sum_{v \in \text{Hamm}} (-1)^{\text{wt}(v)} |v\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle_F - |1\rangle_F \right)$$