classmate

sooraj.ss1729@gmail.com +91-9400635788

INDEX

(10·1)

NAME: SOORAJ·S STD.: SEC.: ROLL NO.: SUB.:

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
		QUANTUM COMPUTATION & QUANTUM INFORMATION – Nielsen & Chuang		

10 Quantum Error Correction

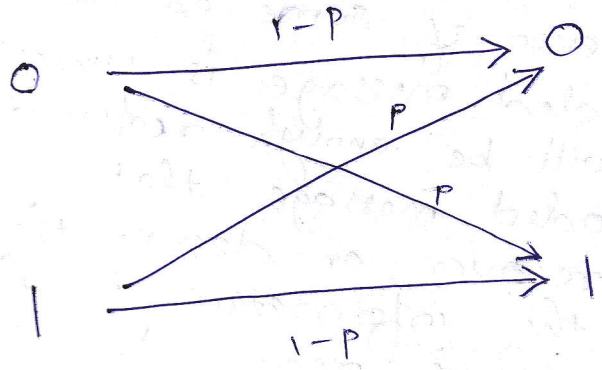
The key idea about error correcting is that,

If we wish to protect a message against the effects of noise, then we should encode the message by adding some redundant information to the message. That way, even if some of the information in the encoded message is corrupted by noise, there will be enough redundancy in the encoded message that it is possible to recover or decode the message so that all the information in the original message is recovered.

Example :

Suppose,
we wish to send a bit from one location to another thro' a noisy classical communication channel.

The effect of the noise in the channel is to flip the bit being transmitted with probability $p > 0$, while with probability $1-p$ the bit is transmitted without error. Such a channel is known as a Binary symmetric channel.



A simple means of protecting the bit against the effects of noise in the binary symmetric channel is to replace the bit we wish to protect with 3 copies of itself:

$$0 \longrightarrow 000$$

$$1 \longrightarrow 111$$

The bit strings 000 and 111 are sometimes referred to as the logical 0 and logical 1, since they play the role of 0 and 1, respectively.

We now send all 3 bits thro' the channel At the receiver's end of the channel 3 bits are output, and the receiver has to decide what the value of the original bit was. Suppose,

001 were output from the channel. Provided the probability p of a bit flip is not too high, it is very likely that the 3rd bit was flipped by the channel, and that 0 was the bit that was sent.

This type of decoding is called majority voting, since the decoded o/p from the channel is whatever value, 0 or 1, appears more times in the actual channel o/p. Majority voting fails if 2 or more of the bits sent thro' the channel were flipped, and succeeds otherwise.

Probability of a bit flip $= 3$ — case 1

Probability that 2 or more of the bits are flipped $= P(2 \text{ bit flips}) + P(3 \text{ bit flips})$ — case 2

$$= 3p^2(1-p) + p^3 = \underline{3p^2 - 2p^3}$$

$\therefore$ The probability of error, $P_e = 3p^2 - 2p^3$ — case

Without encoding, the probability of an error was p; so the code makes the transmission more reliable provided $P_e < p$, which occurs whenever $p < 1/2$.

$P_e < p \Longrightarrow 3p^2 - 2p^3 < p$

$2p^3 - 3p^2 + p = p(2p^2 - 3p + 1) > 0$

$p = \frac{3 \pm 1}{4} = 1 \text{ (or) } 1/2$

$p(p - 1/2)(p - 1) > 0$

Case 1 $p > 0$ & $p - 1/2 > 0$ & $p - 1 > 0$

$p > 0$ & $p > 1/2$ & $p > 1 \Longrightarrow p > 1$ Not possible

Case 2 $p - 1/2 > 0$ & $p < 0$ & $p - 1 < 0$

$p > 1/2$ & $p < 0$ & $p < 1$

$p > 1/2$ & $p < 0 \Longrightarrow$ Not possible p.

Case 3 $p > 0$ & $p - 1/2 < 0$ & $p - 1 < 0$

$p > 0$ & $p < 1/2$ & $p < 1$

$p > 0$ & $p < 1/2$

$\Longrightarrow \underline{p < 1/2}$

The type of code just described is called a repetition code, since we encode the message to be sent by repeating it a # of times.

The 3 qubit bit flip code

We have 3 formidable difficulties to develop quantum error-correcting codes to protect quantum states against the effects of noise,

(i) No cloning – Implementing the repetition code quantum mechanically by duplicating the quantum state 3 or more times, is forbidden by the no-cloning theorem. Even if cloning were possible, it could not be possible to measure & compare the 3 quantum states o/p from the channel.

(ii) Errors are continuous – A continuum of different errors may occur on a single qubit. Determining which error occurred in order to correct it would appear to require infinite precision, and therefore infinite resources.

(iii) Measurement destroys quantum information -

In classical error-correction we observe the o/p from the channel, and decide what decoding procedure to adopt. Observation in quantum mechanics generally destroys the quantum state under observation, and makes recovery impossible.

Suppose,

we send qubits thro' a channel which leaves the qubits untouched with probability $1-p$, and flips the qubits with probability p.

ie.,

with probability p the state $|\psi\rangle$ is taken to the state $X|\psi\rangle$, where X is the usual Pauli σ_x operator, or bit flip operator. This channel is called the bit flip channel.

Bit flip code

Suppose,

we encode the single qubit state $a|0\rangle+b|1\rangle$ in 3 qubits as $a|000\rangle+b|111\rangle$.

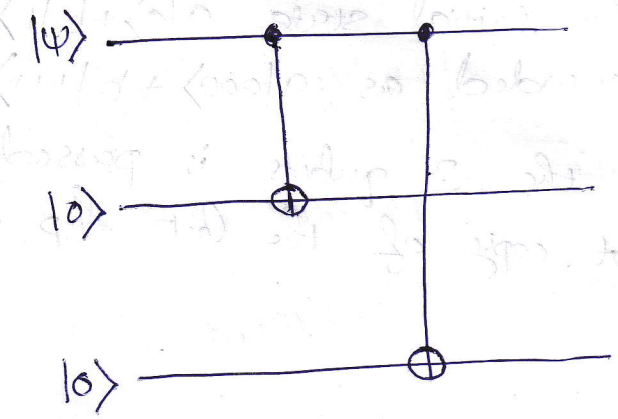
$$|0\rangle \longrightarrow |0_L\rangle \equiv |000\rangle$$

$$|1\rangle \longrightarrow |1_L\rangle \equiv |111\rangle$$

where the superpositions of basis states are taken to corresponding superpositions of encoded states.

The notation $|0_L\rangle$ and $|1_L\rangle$ indicates that these are the logical $|0\rangle$ and logical $|1\rangle$ states, not the physical zero and one states.

A circuit performing this encoding is given as,



* Encoding circuit for the 3 qubit bit flip code

$$|\psi_1\rangle = |\psi\rangle \otimes |0\rangle \otimes |0\rangle = (a|0\rangle + b|1\rangle) \otimes |0\rangle \otimes |0\rangle$$
$$= a|0\rangle \otimes |0\rangle \otimes |0\rangle + b|1\rangle \otimes |0\rangle \otimes |0\rangle$$

$$|\psi_2\rangle = a|0\rangle \otimes |0\rangle \otimes |0\rangle + b|1\rangle \otimes X|0\rangle \otimes |0\rangle$$
$$= a|0\rangle \otimes |0\rangle \otimes |0\rangle + b|1\rangle \otimes |1\rangle \otimes |0\rangle$$

$$|\psi_3\rangle = a|0\rangle \otimes |0\rangle \otimes |0\rangle + b|1\rangle \otimes |1\rangle \otimes X|0\rangle$$
$$= a|0\rangle \otimes |0\rangle \otimes |0\rangle + b|1\rangle \otimes |1\rangle \otimes |1\rangle$$
$$= a|000\rangle + b|111\rangle$$

Now,

Suppose the initial state $a|0\rangle + b|1\rangle$ has been perfectly encoded as $a|000\rangle + b|111\rangle$.
Each of the 3 qubits is passed through an independent copy of the bit flip channel.

Suppose a bit flip occurred on one or fewer of the qubits. There is a simple 2 stage error-correction procedure which can be used to recover the correct quantum state in this case:

Stage 1 : Error detection (or) syndrome diagnosis

We perform a measurement which tells us what error, if any, occurred on the quantum state. The measurement result is called the error syndrome.

For the bit flip channel there are 4 error syndromes, corresponding to the 4 projection operators :

$$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111| \quad \text{no error}$$

$$P_1 \equiv |100\rangle\langle 100| + |011\rangle\langle 011| \quad \text{bit flip on qubit one}$$

$$P_2 \equiv |010\rangle\langle 010| + |101\rangle\langle 101| \quad \text{bit flip on qubit two}$$

$$P_3 \equiv |001\rangle\langle 001| + |110\rangle\langle 110| \quad \text{bit flip on qubit three}$$

Suppose for example that a bit flip occurs on qubit one, so the corrupted state is $a|100\rangle + b|011\rangle$.

$\Rightarrow \langle\psi|P_1|\psi\rangle = 1$

$\Rightarrow$ the outcome of the measurement result (the error syndrome) is certainly 1.

The syndrome measurement does not cause any change to the state : it is $a|100\rangle + b|011\rangle$ both before and after syndrome measurement.

The syndrome contains only information about what error has occurred, and does not allow us to infer anything about the value of a or b, ie., it contains no information about the state being protected. This is a generic feature of syndrome measurements, since to obtain information about the identity of a quantum state it is in general necessary to perturb that state.

Stage 2: We use the value of the error syndrome to tell us what procedure to use to recover the initial state.

Ex:- If the error syndrome was 1, indicating a bit flip on the 1st qubit, then we flip that qubit again, recovering the original state $a|000\rangle + b|111\rangle$ with perfect accuracy.

The 4 possible error syndromes and the recovery procedure in each case are:

0 (no error) – do nothing

1 (Bit flip on 1st qubit) – flip the 1st qubit again

2 (Bit flip on 2nd qubit) – flip the 2nd qubit again

3 (Bit flip on 3rd qubit) – flip the 3rd qubit again

For each value of the error syndrome it is easy to see that the original state is recovered with perfect accuracy, given that the corresp. error occurred.

This error-correction procedure works perfectly, provided bit flip occurs on one or fewer of the 3 qubits. This occurs with probability,

$$= P(\text{No bit flip}) + P(\text{bit flip on one qubit})$$

$$= (1-p)^3 + 3p(1-p)^2$$

$$= (1-p)^2(1-p+3p)$$

$$= (1-p)^2(1+2p)$$

$$= (1-2p+p^2)(1+2p)$$

$$= 1+2p-2p-4p^2+p^2+2p^3$$

$$= 1-3p^2+2p^3$$

The probability of an error remaining uncorrected is $= 1-(1-3p^2+2p^3)$

$$= 3p^2-2p^3$$

just as for the classical repetition code.

$\Rightarrow$ Provided $p < 1/2$ the encoding and decoding improve the reliability of storage of the quantum state.

□ Improving the error analysis

This error analysis is not completely adequate. The problem is that not all errors and states in quantum mechanics are created equal: quantum states live in a continuous space, so it is possible for some errors to corrupt a state by a tiny amount, while others mess it up completely.

Ex:-

the bit flip 'error' X, which does not affect the state $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ at all, but flips the $|0\rangle$ state so it becomes a $|1\rangle$.

In the former case we would not be worried about a bit flip error occurring, while in the latter case we would obviously be very worried.

To address this problem $\Longrightarrow$ Fidelity

The fidelity b/w a pure and a mixed state is given by,

$$F(|\psi\rangle, \rho) = \sqrt{\langle\psi|\rho|\psi\rangle}$$

which is the square root of the overlap b/w $|\psi\rangle$ and ρ.

The object of quantum error-correction is to increase the fidelity with which quantum information is stored (or communicated) up near the maximum possible fidelity of 1.

Compare: min. fidelity achieved by the 3 qubit bit flip code with the fidelity when no error-correction is performed.

Suppose the quantum state of interest is $|\psi\rangle$.

case 1 - Without using the error-correcting code

The state of the qubit after being sent through the channel is,

$$\rho = (1-p)|\psi\rangle\langle\psi| + p X|\psi\rangle\langle\psi|X$$

The fidelity is given by,

$$F = \sqrt{\langle\psi|\rho|\psi\rangle}$$

$$= \sqrt{\langle\psi|\left((1-p)|\psi\rangle\langle\psi| + p\, X|\psi\rangle\langle\psi|X\right)|\psi\rangle}$$

$$= \sqrt{(1-p) + p\langle\psi|X|\psi\rangle\langle\psi|X|\psi\rangle}$$

$$= \sqrt{(1-p) + p\,|\langle\psi|X|\psi\rangle|^2}$$

The 2nd term under the square root is non-negative, and equal to 0 when $|\psi\rangle = |0\rangle$.

$\Rightarrow$ The minimum fidelity is, $F_{min} = \sqrt{1-p}$

ILA (8)

Case 2 – The 3 qubit error-correcting code is used to protect the state $|\psi\rangle = a|0_L\rangle + b|1_L\rangle$

The quantum state after both the noise and error-correction is,

$$\rho = \left[(1-p)^3 + 3p(1-p)^2\right]|\psi\rangle\langle\psi| + \cdots$$

The omitted terms represent contributions from bit flips on 2 or 3 qubits.

All the omitted terms are +ve operators,

$\langle\psi|A|\psi\rangle \geq 0$ for all $|\psi\rangle$ for a +ve operator A.

ILA (8)

$$\therefore F = \sqrt{\langle\psi|\rho|\psi\rangle} \geq \sqrt{(1-p)^3 + 3p(1-p)^2}$$

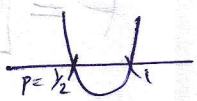
$$= \sqrt{1 - 3p^2 + 2p^3}.$$

ie; The fidelity is at least $\sqrt{1-3p^2+2p^3}$

$$p-(3p^2-2p^3) = 2p^3-3p^2+p = p(2p^2-3p+1) = 0$$

$$\Delta = \sqrt{9-8} = 1$$

$$p = \frac{3 \pm 1}{4} = 1 \text{ (or) } 1/2$$



$p-(3p^2-2p^3) > 0$ for $p < 1/2$

$p > 3p^2-2p^3$

$-p < -3p^2+2p^3 \implies 1-p < 1-3p^2+2p^3$

$\sqrt{1-p} < \sqrt{1-3p^2+2p^3}$ for $p < 1/2$

∴

The fidelity of storage for the quantum state is improved provided $p < 1/2$.

Ex:10.9 The action of the bit flip channel can be described by the quantum operation $\mathcal{E}(\rho) = (1-p)\rho + pX\rho X$. Show that this may be given an alternate operator-sum representation, as $\mathcal{E}(\rho) = (1-2p)\rho + 2pP_+\rho P_+ + 2pP_-\rho P_-$ where P_+ and P_- are projectors onto the +1 and -1 eigenstates of X, $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$, respectively. This later representation can be understood as a model in which the qubit is left alone with probability $1-2p$, and is 'measured' by the environment in the $|+\rangle, |-\rangle$ basis with probability $2p$.

Ans: $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ has eigenvalues & eigenvectors:

$$\lambda_- = -1 \quad , \quad |\lambda_-\rangle = |-\rangle$$

$$\lambda_+ = +1 \quad , \quad |\lambda_+\rangle = |+\rangle$$

$$X = |+\rangle\langle+| - |-\rangle\langle-| = P_+ - P_-$$

$$I = |+\rangle\langle+| + |-\rangle\langle-| = P_+ + P_-$$

$$\mathcal{E}(\rho) = (1-p)\rho + pX\rho X$$

$$= (1-2p)\rho + p\rho + pX\rho X$$

$$= (1-2p)\rho + pX\rho X + pI\rho I$$

$$= (1-2p)\rho + p(P_+ - P_-)\rho(P_+ - P_-)$$

$$+ p(P_+ + P_-)\rho(P_+ + P_-)$$

$$= (1-2p)\rho + 2pP_+\rho P_+ + 2pP_-\rho P_-$$

This is equivalent to measuring the state in the $|+\rangle, |-\rangle$ state with probability $2p$.

Syndrome measurement — another way

Suppose that instead of measuring the 4 projectors P_0, P_1, P_2, P_3, we performed two measurements —

1st: the observable $Z_1 Z_2$ (ie; $Z \otimes Z \otimes I$)

2nd: the observable $Z_2 Z_3$ (ie; $I \otimes Z \otimes Z$)

$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \Rightarrow \lambda_\pm = \pm 1, \; |\lambda_+\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; |\lambda_-\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Each of these observables has eigenvalues ± 1.

∴

Each measurement provides a single bit of information, for a total of 2 bits of information — 4 possible syndromes, just as in the earlier description.

The 1st measurement, of $Z_1 Z_2$, can be thought of as comparing the 1st and 2nd qubits to see if they are the same.

$$Z_1 Z_2 = Z \otimes Z \otimes I = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \otimes I$$

$$= (|0\rangle\langle 0| - |1\rangle\langle 1|) \otimes (|0\rangle\langle 0| - |1\rangle\langle 1|) \otimes I$$

$$= (|0\rangle\langle 0| \otimes |0\rangle\langle 0| + |1\rangle\langle 1| \otimes |1\rangle\langle 1| - |0\rangle\langle 0| \otimes |1\rangle\langle 1| - |1\rangle\langle 1| \otimes |0\rangle\langle 0|) \otimes I$$

$$= (|00\rangle\langle 00| + |11\rangle\langle 11|) \otimes I - (|01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I$$

which corresp. to a projective measurement with projectors $(|00\rangle\langle 00| + |11\rangle\langle 11|) \otimes I$ and $(|01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I$.

$\Rightarrow$ Measuring $Z_1 Z_2$ can be thought of as comparing the values of the 1st and 2nd qubits, giving +1 if they are the same, and -1 if they are different.

Similarly,

measuring Z_2Z_3 compares the values of the 2nd and 3rd qubits, giving +1 if they are the same, and -1 if they are different.

Combining these 2 measurements results can determine whether a bit flip occurred on one of the qubits or not, and if so, which one:

if both measurement results give +1 then with high probability no bit flip has occurred;

if measuring Z_1Z_2 gives +1 and measuring Z_2Z_3 gives -1 then with high probability just the 3rd qubit flipped;

if measuring Z_1Z_2 gives -1 and measuring Z_2Z_3 gives +1 then with high probability just the 1st qubit flipped;

Finally, if both measurements give -1 then with high probability just the 2nd qubit flipped.

What's crucial to the success of these measurements is that neither measurement gives any information about the amplitudes 'a' and 'b' of the encoded quantum state, and thus neither measurement destroys the superpositions of quantum states that we wish to preserve using the code.

Ex: 10.3

Ans:

Ex: 10.3 Show by explicit calculation that measuring Z_1Z_2 followed by Z_2Z_3 is equivalent, up to labeling of the measurement outcomes, to measuring the 4 projectors in the sense that both procedures result in the same measurement statistics and post-measurement states.

Ans:

□ 3 qubit phase flip code

The bit flip code is interesting, but it does not appear to be that significant an innovation over classical error-correcting codes, and leaves many problems open (Ex:- many kinds of errors other than bit flips can happen to qubits).

A more interesting noisy quantum channel is the phase flip error model for a single qubit

- the qubit is left alone with probability $1-p$, and with probability p the relative phases of the $|0\rangle$ and $|1\rangle$ states is flipped.

ie., the phase flip operator Z is applied to the qubit with probability $p>0$, so that the state $a|0\rangle+b|1\rangle$ is taken to the state $a|0\rangle-b|1\rangle$ under the phase flip.

There is no classical equivalent to the phase flip channel, since classical channels don't have any property equivalent to phase.

Phase flip channel $\longrightarrow$ bit flip channel.

Suppose we work in the qubit basis $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. W.r.t this basis the operator Z takes $|+\rangle$ to $|-\rangle$ and vice versa,

$Z|0\rangle = |0\rangle$
$Z|1\rangle = -|1\rangle$

i.e., it acts just like a bit flip w.r.t the labels $+$ to $-$!

$\Rightarrow$ This suggests using the states $|0_L\rangle = |+++\rangle$ and $|1_L\rangle = |---\rangle$ as logical zero and one states for protection against phase flip errors.

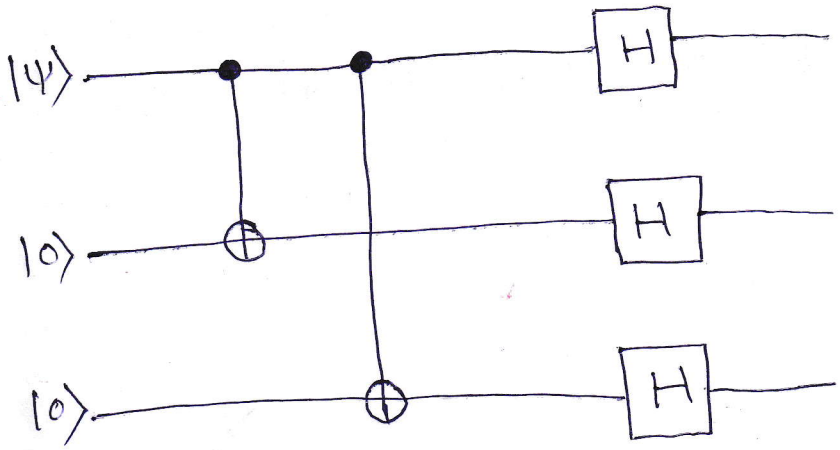
All the operations needed for error-correction encoding, error-detection and recovery - are performed just as for the bit flip channel, but wrt the $|+\rangle, |-\rangle$ basis instead of the $|0\rangle, |1\rangle$ basis.

To accomplish this basis change we simply apply the Hadamard gate and its inverse (also the Hadamard gate) at appropriate points in the procedure.

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

&

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

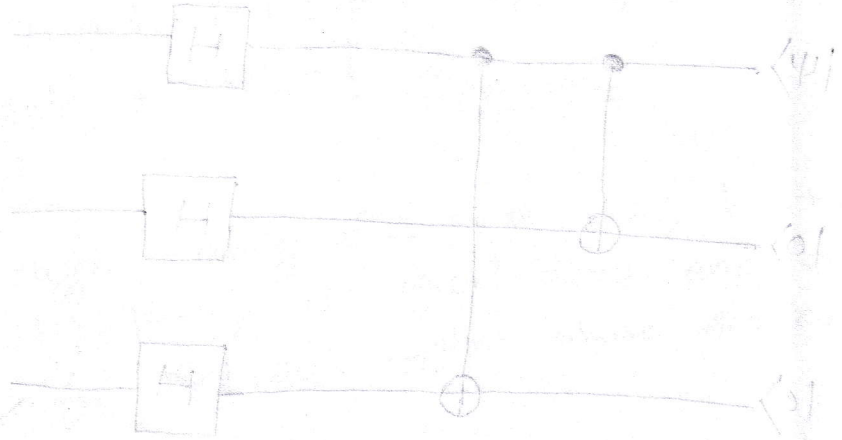


* Encoding circuits for the phase flip code.

Encoding: 1st we encode in 3 qubits exactly as for the bit flip channel and 2nd apply a Hadamard gate to each qubit.

Error detection: achieved by applying the same projective measurements as in the bit flip case, but conjugated by Hadamard gates -

$$P_j \longrightarrow P_j' \equiv H^{\otimes 3} P_j H^{\otimes 3}$$



Equivalently, syndrome measurement may be performed by measuring the observables

$$H^{\otimes 3} Z_1 Z_2 H^{\otimes 3} = (H \otimes H \otimes H)(Z \otimes Z \otimes I)(H \otimes H \otimes H)$$

$$= HZH \otimes HZH \otimes HIH$$

$$= X \otimes X \otimes I = X_1 X_2$$

and $H^{\otimes 3} Z_2 Z_3 H^{\otimes 3} = X_2 X_3$.

Measurement of the observables $X_1 X_2$ and $X_2 X_3$ corresponds to comparing the sign of the qubits one and two, and two and three, respectively, in the sense that measurement of $X_1 X_2$ gives $+1$ for states like $|+\rangle|+\rangle \otimes (\cdot)$ or $|-\rangle|-\rangle \otimes (\cdot)$, and -1 for states like $|+\rangle|-\rangle \otimes (\cdot)$ or $|-\rangle|+\rangle \otimes (\cdot)$.

Recovery : Hadamard-conjugated recovery operation from the bit flip code

Ex:-

Suppose we detected a flip in the sign of the 1st qubit from $|+\rangle$ to $|-\rangle$.

Then we recover by applying $HX_1H = Z_1$ to the 1st qubit.

This code for the phase flip channel has the same characteristics as the code for the bit flip channel. In particular, the minimum fidelity for the phase flip code is the same as that for the bit flip code, and we have the same criteria for the code producing an improvement over the case with no error-correction.

⇒ These 2 channels are unitarily equivalent, since there is a unitary operator U (in this case the Hadamard gate) such that the action of one channel is the same as the other, provided the 1st channel is preceded by U and followed by $U^\dagger$.

These operations may be trivially incorporated into the encoding & error-correction operations.

Ex: 10.4 Consider the 3 qubit bit flip code.

Suppose we had performed the error syndrome measurement by measuring the 8 orthogonal projectors corresp. to projections onto 8 computational basis states.

(a) Write out the projectors corresp. to this measurement and explain how the measurement result can be used to diagnose the error syndrome: either no bits flipped or bit # j flipped where $j = 1, 2, 3$.

(b) Show that the recovery procedure works only for computational basis states.

(c) What's the min. fidelity for the error-correction procedure?

The Shor code

- simple quantum code which can protect against the effects of an arbitrary error on a single qubit.

- combination of 3 qubit phase flip and bit flip codes.

1st : encode each of these qubits using the 3 qubit phase flip code

$$|0\rangle \longrightarrow |+++\rangle \,, \quad |1\rangle \longrightarrow |---\rangle$$

2nd : encode each of these qubits using the 3 qubit bit flip code

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \longrightarrow \frac{|000\rangle + |111\rangle}{\sqrt{2}}$$

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \longrightarrow \frac{|000\rangle - |111\rangle}{\sqrt{2}}$$

The result is a 9 qubit code, with codewords given by,

$$|0\rangle \rightarrow |0_L\rangle = \frac{(|000\rangle+|111\rangle)\otimes(|000\rangle+|111\rangle)\otimes(|000\rangle+|111\rangle)}{2\sqrt{2}}$$

$$|1\rangle \rightarrow |1_L\rangle = \frac{(|000\rangle-|111\rangle)\otimes(|000\rangle-|111\rangle)\otimes(|000\rangle-|111\rangle)}{2\sqrt{2}}$$

$$\alpha|0\rangle+\beta|1\rangle \rightarrow \alpha|0_L\rangle+\beta|1_L\rangle =$$

$$= \frac{\alpha}{2\sqrt{2}}(|000\rangle+|111\rangle)\otimes(|000\rangle+|111\rangle)\otimes(|000\rangle+|111\rangle)$$

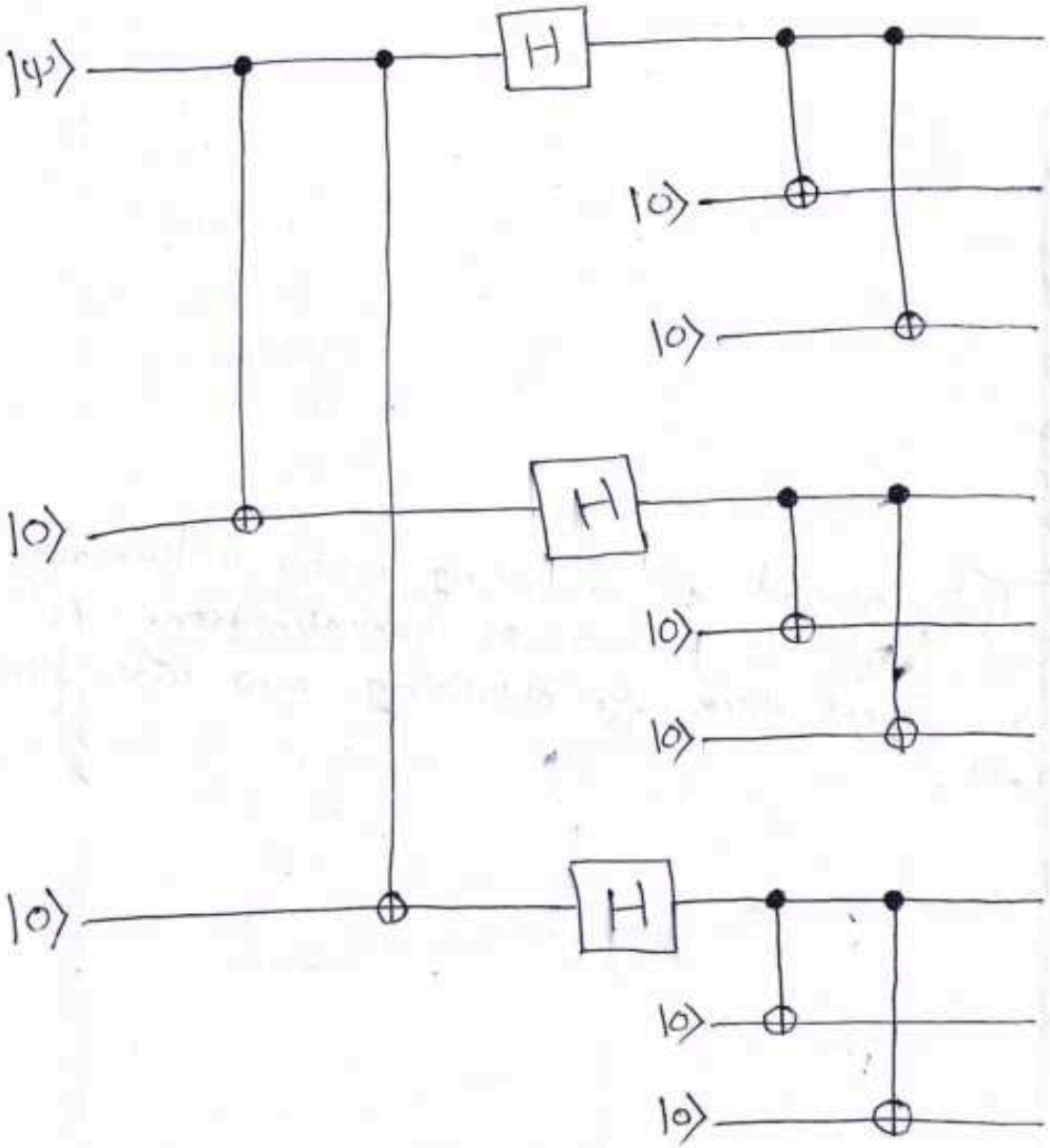
$$+ \frac{\beta}{2\sqrt{2}}(|000\rangle-|111\rangle)\otimes(|000\rangle-|111\rangle)\otimes(|000\rangle-|111\rangle)$$

$|\psi\rangle$ —

$|0\rangle$ —

$|0\rangle$ —

*



* Encoding circuit for the Shor 9 qubit code.

- The Shor code is able to protect against phase flip and bit flip errors on any qubit.

---

bit flip

Suppose a bit flip occurs on the 1st qubit. As for the bit flip code, we perform a measurement of Z_1Z_2 comparing the 1st two qubits, and find that they are different. We conclude that a bit flip error occurred on the 1st or 2nd qubit. Next, we compare the 2nd and 3rd qubit by performing a measurement of Z_2Z_3. We find that they are the same, so it could not have been the 2nd qubit which flipped.

$\Rightarrow$ the 1st qubit must have flipped, and recover from the error by flipping the 1st qubit again, back to its original state.

In a similar way, we can detect and recover from the effects of bit-flip errors on any 9 qubits in the code.

ie.,
Measure parities with 3 qubit Blocks: Z_1Z_2, Z_2Z_3; Z_4Z_5, Z_5Z_6; Z_7Z_8, Z_8Z_9.
If X error is detected in i^{th} qubit, correct by gate X_i.
(If bit-flips in different 3-qubit Blocks, they all can be corrected)

Phase flip

Suppose a phase flip occurs on the 1st qubit.

Such a phase flip flips the sign of the 1st block of qubits, changing $|000\rangle+|111\rangle$ to $|000\rangle-|111\rangle$, and vice versa.

The state becomes,

$$\frac{\alpha}{2\sqrt{2}}\left(|000\rangle-|111\rangle\right)\otimes\left(|000\rangle+|111\rangle\right)\otimes\left(|000\rangle+|111\rangle\right)$$
$$+\frac{\beta}{2\sqrt{2}}\left(|000\rangle+|111\rangle\right)\otimes\left(|000\rangle-|111\rangle\right)\otimes\left(|000\rangle-|111\rangle\right)$$

A phase flip on any of the 1st three qubits has this effect.

The key idea here is to detect which of the 3 blocks of 3 qubits has experienced a change of sign.

Syndrome measurement begins by comparing the sign of the 1st and 2nd blocks of 3 qubits.

For example, $(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle)$ has the sign (+) in both blocks of qubits, while $(|000\rangle - |111\rangle) \otimes (|000\rangle + |111\rangle)$ has different signs.

When a phase flip occurs on any of the 1st three qubits, we find that the signs of the 1st and 2nd blocks are different.

The 2nd and final stage of syndrome measurement is to compare the sign of the 2nd and 3rd blocks of qubits. We find that these are the same, and conclude that the phase must have flipped in the 1st block of 3 qubits.

We recover from this by flipping the sign in the 1st block of 3 qubits back to its original value.

We can recover from a phase flip on any of the 9 qubits in a similar manner.

ie.,

Measure parities of phases of 3 qubit blocks:

$X_1X_2X_3X_4X_5X_6$, $X_4X_5X_6X_7X_8X_9$

($X_1X_2X_3$ changes sign of superposition if $000 - 111$ and does nothing if $000 + 111$.

If phase flip error is detected in j^{th} 3-qubit block, correct by applying Z-gate to any qubit in this block.

Note: This step is insensitive to bit flips

Ex:-

$$X_1X_2X_3(|001\rangle - |110\rangle) = |110\rangle - |001\rangle = -(|001\rangle - |110\rangle)$$

⇒ We can first detect errors, then correct.

bit & phase flip

Suppose both bit and phase flip errors occur on the 1st qubit, ie., the operator ZX_1 is applied to that qubit.

The procedure for detecting a bit flip error will detect a bit flip on the 1st qubit, and correct it, and the procedure for detecting a phase flip error will detect a phase flip on the 1st Block of 3 qubits, and correct it.

bit & phase flip

Suppose both bit and phase flip errors occur on the 1st qubit, ie., the operator $Z_1 X_1$ is applied to that qubit.

The procedure for detecting a bit flip error will detect a bit flip on the 1st qubit, and correct it, and the procedure for detecting a phase flip error will detect a phase flip on the 1st Block of 3 qubits, and correct it.

Ex: 10.5

Show that the syndrome measurement for detecting phase flip errors in the Shor code corresponds to measuring the observables $X_1X_2X_3X_4X_5X_6$ and $X_4X_5X_6X_7X_8X_9$.

Ex: 10.6 Show that recovery from a phase flip on any of the 1st 3 qubits may be accomplished by applying the operator $Z_1Z_2Z_3$.

Shor code — arbitrary errors

Shor code protects against completely arbitrary errors, provided they only affect a single qubit!

The error can be tiny - a rotation about the x-axis of the Bloch sphere by $\pi/263$ radians, say, or it can be an apparently disastrous error like removing the qubit entirely and replacing it with garbage!

This is an extraordinary fact that the apparent continuum of errors that may occur on a single qubit can all be corrected by correcting only a discrete subset of those errors; all other possible errors being corrected automatically by this procedure!

Suppose noise of an arbitrary type is occurring on the i^{st} qubit only.

We describe noise by a trace-preserving quantum operation $\mathcal{E}$.

Analyze error-correction by expanding $\mathcal{E}$ in an operator-sum representation with operation elements $\{E_i\}$.

Suppose,
the state of the encoded qubit is $|\psi\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$ before the noise acts, then after the noise has acted the state is,

$$\mathcal{E}(|\psi\rangle\langle\psi|) = \sum_i E_i |\psi\rangle\langle\psi| E_i^\dagger$$

Focus on the effect error-correction has on a single term in this sum, say $E_i|\psi\rangle\langle\psi|E_i^\dagger$.

E_i is an operator on the 1st qubit alone, which can be expanded in the Pauli matrix basis,

ie, expanded as a linear combination of the identity I, the bit flip X_1, the phase flip Z_1, and the combined bit and phase flip $-iY=X_1Z_1$:

$$E_i = e_{i0} I + e_{i1} X_1 + e_{i2} Z_1 + e_{i3} X_1 Z_1$$

The (un-normalized) quantum state $E_i|\psi\rangle$ can thus be written as a superposition of 4 terms, $|\psi\rangle, X_1|\psi\rangle, Z_1|\psi\rangle, X_1Z_1|\psi\rangle$.

$$E_i|\psi\rangle = e_{i0}|\psi\rangle + e_{i1} X_1|\psi\rangle + e_{i2} Z_1|\psi\rangle + e_{i3} X_1 Z_1|\psi\rangle$$

Measuring the error syndrome collapses this superposition into one of the 4 states: $|\psi\rangle$, $X_1|\psi\rangle$, $Z_1|\psi\rangle$ or $X_1 Z_1|\psi\rangle$ from which recovery may then be performed by applying the appropriate inversion operation, resulting in the final state $|\psi\rangle$.

The same is true for all the other operation elements E_i.

Thus,
error-correction results in the original state $|\psi\rangle$ being recovered, despite the fact that the error on the 1st qubit was arbitrary.

⇒ A fundamental & deep fact about quantum error-correction

- by correcting just a discrete set of errors, the bit flip, phase flip and combined bit-phase flip, in this example, a quantum error-correcting code is able to automatically correct an apparently larger (continuous!) class of errors.

the original
he fact that
was arbit

discrete set of
nd combined
example, a
ting code is
ct an appare
ars of errors

10.3 Theory of quantum error-correction

Quantum states are encoded by a unitary operation into a *quantum error-correcting code*, formally defined as a subspace C of some larger Hilbert space. It is useful to have a notation for the projector onto the code space C, so we use the notation P; for the three qubit bit flip code $P = |000\rangle\langle 000| + |111\rangle\langle 111|$.

→ code subspace

ie.,

The encoded quantum information is stored in a specific region of the larger Hilbert space called the code subspace C.

Ex:- In the bit flip code, the logical states $|0_L\rangle$ and $|1_L\rangle$ are encoded as: $|0_L\rangle = |000\rangle$ & $|1_L\rangle = |111\rangle$

and the code subspace C can be represented by a projector P, s.t, for the bit flip code, the projector is $P = |000\rangle\langle 000| + |111\rangle\langle 111|$, which identifies states in the subspace C.

After encoding, the quantum state is subjected to noise, which moves the state outside the codespace C. A measurement called the syndrome measurement is then performed to diagnose the type of error; without destroying the encoded state.

Each error maps the state into a subspace that is distinct from others, which ensures the syndrome measurement can reliably distinguish b/w errors.

Different error syndromes corresp. to orthogonal subspaces of the total Hilbert space.

The subspace must remain undeformed versions of the original code space, so errors must map codewords to orthogonal states.

To develop a general theory of quantum error correction, we assume:

- Noise is described by a quantum operation $\mathcal{E}$
- The error-correction process is performed by a trace-preserving quantum operation $\mathcal{R}$, called the error-correction operation. $\mathcal{R}$ combines both error detection and recovery into a single quantum operation.

In order for error-correction to be deemed successful, we require that for any state ρ whose support lies in the code C,

$$(\mathcal{R} \circ \mathcal{E})(\rho) \propto \rho. \tag{10.15}$$

The quantum error correction is only defined for states within the code subspace C. These are the states we want to protect and recover after errors.

ie: the condition $(\mathcal{R} \circ \mathcal{E})(\rho) \propto \rho$ applies to states ρ that are encoded in the code subspace C. If a state is disturbed by noise $\mathcal{E}$, it may temporarily leave C. The recovery operation $\mathcal{R}$ must map it back into C, restoring it.

You may be wondering why we wrote $\propto$ rather than $=$ in the last equation. If $\mathcal{E}$ were a trace-preserving quantum operation then by taking traces of both sides of the equation we see that $\propto$ would be $=$. However, sometimes we may be interested in error-correcting non-trace-preserving operations $\mathcal{E}$, such as measurements, for which $\propto$ is appropriate. Of course, the error-correction step $\mathcal{R}$ must succeed with probability 1, which is why we required $\mathcal{R}$ to be trace-preserving.

The *quantum error-correction conditions* are a simple set of equations which can be checked to determine whether a quantum error-correcting code protects against a particular type of noise $\mathcal{E}$. We will use these conditions to construct a plethora of quantum codes, and also to investigate some of the general properties of quantum error-correcting codes.

Theorem 10.1: (**Quantum error-correction conditions**) Let C be a quantum code, and let P be the projector onto C. Suppose $\mathcal{E}$ is a quantum operation with operation elements $\{E_i\}$. A necessary and sufficient condition for the existence of an error-correction operation $\mathcal{R}$ correcting $\mathcal{E}$ on C is that

$$PE_i^\dagger E_j P = \alpha_{ij} P, \tag{10.16}$$

for some Hermitian matrix α of complex numbers.

We call the operation elements $\{E_i\}$ for the noise $\mathcal{E}$ *errors*, and if such an $\mathcal{R}$ exists we say that $\{E_i\}$ constitutes a *correctable set of errors*.

Proof (sufficiency condition)

Suppose $\{E_i\}$ is a set of operation elements satisfying the quantum error correction conditions in 10.16.

By assumption, α is a Hermitian matrix

$\Rightarrow \alpha = u d u^\dagger$ (or) $d = u^\dagger \alpha u$

where u: unitary matrix and d: diagonal matrix.

Define operators, $F_k \equiv \sum_i u_{ik} E_i$

Theorem 8.2 (QC-13) $\Rightarrow \{F_k\}$ is also a set of operation elements for $\mathcal{E}$

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger = \sum_k F_k \rho F_k^\dagger$$

$\therefore \quad P F_k^{\dagger} F_l P = \sum u$

10.4 Constructing quantum codes

Classical Linear Codes

A *linear code* C encoding k bits of information into an n bit code space is specified by an n by k *generator matrix* G whose entries are all elements of $\mathbf{Z}_2$, that is, zeroes and ones. The matrix G maps messages to their encoded equivalent. Thus the k bit message x is encoded as Gx, where the message x is treated as a column vector in the obvious way. Furthermore, the multiplication operation, and all our other arithmetic operations in this section, are done modulo 2.

Ex: the repetition code mapping a single bit to 3 repetitions is specified by the generator matrix, $G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$.

G maps the possible messages, 0 and 1, to their encoded form $G[0] = (0,0,0)$ and $G[1] = (1,1,1)$

a code using n bits to encode k bits of information is an $[n, k]$ code

→ a $[3,1]$ code.

Ex: encode 2 bits using 3 repetions of each bit — a $[6,2]$ code

The generator matrix $G = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$ such that

$G(0,0) = (0,0,0,0,0,0)$; $G(0,1) = (0,0,0,1,1,1)$

$G(1,0) = (1,1,1,0,0,0)$; $G(1,1) = (1,1,1,1,1,1)$

just as we expect. The set of possible codewords for the code corresponds to the vector space spanned by the columns of G, so in order that all messages be uniquely encoded we require that the columns of G be linearly independent, but otherwise place no constraints on G.

Exercise 10.14: Write an expression for a generator matrix encoding k bits using r repetitions for each bit. This is an $[rk, k]$ linear code, and should have an $rk \times k$ generator matrix.

Exercise 10.15: Show that adding one column of G to another results in a generator matrix generating the same code.

A great advantage of linear codes over general error-correcting codes is their compact specification.

Encoding k bits in n bits

General code: code takes a message of k-bits & maps it into an n-bit codeword.

For k-bit messages, there are 2^k possible combinations of input messages.

∴ To fully describe the encoding process, we need to specify each of the 2^k codewords, each of which is n-bits long.

∴ The total information needed to describe the encoding is:

# of codewords × length of each codeword = $2^k \times n$

Linear code: only requires specifying the $n \times k$ generator matrix G.

⇒ only nk bits are needed

⇒ exponential reduction in memory requirements.

Encoding - Multiplies the k-bit message by $n \times k$ generator matrix G to obtain n bit encoded message

⟹ $O(nk)$ operations.

Error correction for linear codes ?

$\Longrightarrow$ Parity check matrices

(alternative but equivalent formulation of linear codes)

of linear codes in terms of *parity check* matrices. In this definition an $[n, k]$ code is defined to consist of all n-element vectors x over $\mathbf{Z}_2$ such that

$$Hx = 0, \tag{10.57}$$

where H is an $n - k$ by n matrix known as the *parity check matrix*, with entries all zeroes and ones. Equivalently, but more succinctly, the code is defined to be the kernel of H. A code encoding k bits has 2^k possible codewords so the kernel of H must be k-dimensional, and therefore we require that H have linearly independent rows.

ie., $C = \{x \in \mathbb{Z}_2^n : Hx = 0\} = \ker(H)$

All 2^k valid codewords form a subspace of dimension k within $\mathbb{Z}_2^n$, ie., $\ker(H)$

$\dim(\ker(H)) = n - \dim(\text{rowspace}(H))$

$k = n - \#(\text{independent rows of } H)$

$\Longrightarrow$ H have $n-k$ linearly independent rows.

Choosing $H_{(n-k)\times n}$: H have linearly independent rows (all rows)

- Adding one row of the parity check matrix to another does not change the code.

- Using Gaussian elimination and swapping of bits it is therefore possible to assume that the parity check matrix has the *standard form* $[A|I_{n-k}]$, where A is an $(n-k)\times k$ matrix.

$$\text{rank}(H) = n-k \implies \text{rref}(H) = \left[I_{n-k} \mid A_{(n-k)\times k} \right]$$

$$\ker(H) = \ker(\text{rref}(H)) \quad \left\{ Hx = 0 \implies \text{rref}(H)\, x = 0 \right.$$

P : column swapping matrix

$$Hx = 0 \implies HPP^{-1}x = 0 \implies H'x' = 0$$

where,

$H' = HP$: parity-check matrix after column swapping

$x' = P^{-1}x$: new codeword after reordering its components

$$\left[I_{n-k} \mid A_{(n-k)\times k} \right] \xrightarrow{\text{swapping of bit positions}} \left[A_{(n-k)\times k} \mid I_{n-k} \right]$$

$H \rightleftharpoons G$?

$G_{n \times k}$ & $H_{(n-k) \times n}$

G : generator matrix, used to encode messages into codewords.

H : parity check matrix, used to check whether a vector is a valid codeword (ie., $Hx = 0$).

Construct G from H :

- codewords x must satisfy $Hx = 0$
  ie., codewords lie in the $\ker(H)$.
- choose k linearly independent vectors $y_1, \ldots, y_k$ that span $\ker(H)$
- $\{y_i\}_{i=1}^{k}$ form the basis for the code space (set of all valid codewords).
- Set G to have $y_1, \ldots, y_k$ as its columns

Exercise 10.18: Show that the parity check matrix H and generator matrix G for the same linear code satisfy $HG = 0$.

By definition, any valid codeword x satisfies $Hx = 0$

The columns of G, say $g_1, g_2, \ldots, g_k$ span the code space.

$\Rightarrow Hg_i = 0$ for any $i = 1, 2, \ldots, k$

$$H_{(n-k)\times n} G_{n\times k} = HG = H\begin{bmatrix} g_1 & \cdots & g_k \end{bmatrix}$$
$$= \begin{bmatrix} Hg_1 & \cdots & Hg_k \end{bmatrix}$$
$$= \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} = 0$$

Constructing H from G:

$$HG = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{n-k} \end{bmatrix} \begin{bmatrix} g_1 & g_2 & \cdots & g_k \end{bmatrix} = \begin{bmatrix} H_1 \cdot g_1 & H_1 \cdot g_2 & \cdots & H_1 \cdot g_k \\ H_2 \cdot g_1 & H_2 \cdot g_2 & \cdots & H_2 \cdot g_k \\ \vdots & \vdots & \ddots & \vdots \\ H_{n-k} \cdot g_1 & H_{n-k} \cdot g_2 & \cdots & H_{n-k} \cdot g_k \end{bmatrix} \quad \text{mod } 2 = 0$$

$\Rightarrow$ Each row of H is orthogonal to all columns of G

i.e.,

The columns of G span the code space, a k-dimensional subspace of $\mathbb{Z}_2^n$.

$\Rightarrow$ rows of H span the orthogonal complement of this subspace (an $(n-k)$-dimensional subspace of $\mathbb{Z}_2^n$).

$\Rightarrow$ pick $(n-k)$ linearly independent vectors $y_1, y_2, \ldots, y_{n-k}$ orthogonal to ($a \cdot b \bmod 2 = 0$) the columns of G, and set the rows of H to be $y_1^T, y_2^T, \ldots, y_{n-k}^T$.

Suppose an $[n, k]$ linear code C has a parity check matrix of the form $H = [A|I_{n-k}]$, for some $(n-k)\times k$ matrix A. Show that the corresponding generator matrix is

$$G = \left[\frac{I_k}{-A}\right]. \tag{10.59}$$

(Note that $-A = A$ since we are working modulo 2; however, this equation also holds for linear codes over more general fields than $\mathbf{Z}_2$.)

Let G has the form $G = \begin{bmatrix} I_k \\ X \end{bmatrix}$ where $X_{(n-k)\times k}$

$$HG = \begin{bmatrix} A & I_{n-k} \end{bmatrix} \begin{bmatrix} I_k \\ X \end{bmatrix} = A I_k + I_{n-k} X = A + X = 0$$

$$\Rightarrow X = -A \qquad -x \equiv x \bmod 2$$

$$\therefore G = \begin{bmatrix} I_k \\ -A \end{bmatrix} = \begin{bmatrix} I_k \\ A \end{bmatrix}$$

$\square$

The parity check matrix makes error-detection and recovery quite transparent. Suppose that we encode the message x as $y = Gx$, but an error e due to noise corrupts y giving the corrupted codeword $y' = y + e$.

$Hy = 0$ $\forall$ codeword y $\Longrightarrow$ $Hy' = He$

Hy': Error syndrome

Suppose no errors or just one error occurred. Then the error syndrome is:

$$Hy' = \begin{cases} 0 & \text{no error} \\ He_j & \text{an error occurs on the } j\text{th bit} \end{cases}$$

where e_j : unit vector with component 1 in the jth component.

If we assume that errors occur on at most one bit, it is therefore possible to perform error-correction by computing the error syndrome Hy' and comparing it to the different possible values of He_j to determine which (if any) bit needs to be corrected.

Suppose x and y are words of n bits each. The *(Hamming) distance* $d(x, y)$ between x and y is defined to be the number of places at which x and y differ. Thus $d((1, 1, 0, 0), (0, 1, 0, 1)) = 2$, for example. The *(Hamming) weight* of a word x is defined to be the distance from the string of all zeroes, $\text{wt}(x) \equiv d(x, 0)$, that is, the number of places at which x is non-zero. Note that

$$d(x,y) = \text{wt}(x+y)$$

To understand the connection with error-correction suppose we encode x as $y = Gx$ using a linear error-correcting code. Noise corrupts the encoded bit string producing $y' = y+e$. Provided the probability of a bit flip is less than $1/2$, the most likely codeword to have been encoded is the codeword y which minimizes the number of bit flips needed to get from y to y', that is, which minimizes $\mathrm{wt}(e) = d(y, y')$. In principle, error-correction with a linear code may be accomplished by simply replacing y' by such a y. In practice this may be rather inefficient, since determining the minimal distance $d(y, y')$ in general requires searching all 2^k possible codewords y. A great deal of effort in classical coding theory has gone into constructing codes with special structure that enable error-correction to be performed more efficiently, however these constructions are beyond the scope of this book.

Example 1: $p < \frac{1}{2}$

Suppose $p = 0.2$. Each bit has a 20% chance of flipping, meaning the bits are more likely to remain unchanged ($1 - p = 0.8$).

Case 1: Original codeword $y_1 = 000$

- Possible received strings y':
  - 000: No flips (probability $0.8^3 = 0.512$),
  - 001: One bit flipped (probability $3 \cdot 0.2 \cdot 0.8^2 = 0.384$),
  - 011: Two bits flipped (probability $3 \cdot 0.2^2 \cdot 0.8 = 0.096$),
  - 111: All bits flipped (probability $0.2^3 = 0.008$).

Observations:

- The most likely string is 000, the original codeword.
- Strings closer to 000 in terms of Hamming distance (e.g., 001) have higher probabilities than strings farther away (e.g., 111).

Decoding:

If $y' = 001$, the algorithm calculates the Hamming distance to both codewords:

$$d(001, 000) = 1, \quad d(001, 111) = 2.$$

Thus, y' is decoded as 000, the correct codeword.

Example 2: $p = \frac{1}{2}$

Now let $p = 0.5$. Each bit has an equal chance of flipping or not flipping.

Case 1: Original codeword $y_1 = 000$

- Possible received strings y':
  - 000: No flips (probability $0.5^3 = 0.125$),
  - 001: One bit flipped (probability $3 \cdot 0.5^3 = 0.375$),
  - 011: Two bits flipped (probability $3 \cdot 0.5^3 = 0.375$),
  - 111: All bits flipped (probability $0.5^3 = 0.125$).

Example 3: $p > \frac{1}{2}$

Now let $p = 0.8$. Each bit is more likely to flip ($1 - p = 0.2$).

Case 1: Original codeword $y_1 = 000$

- Possible received strings y':
  - 000: No flips (probability $0.2^3 = 0.008$),
  - 001: One bit flipped (probability $3 \cdot 0.8 \cdot 0.2^2 = 0.096$),
  - 011: Two bits flipped (probability $3 \cdot 0.8^2 \cdot 0.2 = 0.384$),
  - 111: All bits flipped (probability $0.8^3 = 0.512$).

Observations:

- The most likely string is 111, not 000.
- The noise overwhelms the signal, and 111 is incorrectly interpreted as the original codeword.

Decoding:

If $y' = 111$, the algorithm calculates:

$$d(111, 000) = 3, \quad d(111, 111) = 0.$$

Thus, y' is decoded as 111, which is incorrect.

We define the *distance* of a code to be the minimum distance between any two codewords,

$$d(C) \equiv \min_{x,y \in C, x \neq y} d(x,y). \tag{10.60}$$

But $d(x,y) = \mathrm{wt}(x+y)$. Since the code is linear, $x+y$ is a codeword if x and y are, so we see that

C is a subspace.
$c_1, c_2 \in C \Rightarrow c_1 + c_2 \in C$

$$d(C) = \min_{x \in C, x \neq 0} \mathrm{wt}(x). \tag{10.61}$$

Setting $d \equiv d(C)$, we say that C is an $[n,k,d]$ code. The importance of the distance is that a code with distance at least $2t+1$ for some integer t is able to correct errors on up to t bits, simply by decoding the corrupted encoded message y' as the unique codeword y satisfying $d(y,y') \leq t$.

Proof.

Let y : transmitted codeword, y' : received codeword s.t $d(y,y') \leq t$

The decoding algorithm chooses the codeword y closest to y'.

Triangle inequality $\Rightarrow$ $d(y,y') + d(y',y_2) \geq d(y,y_2)$

$$d(y',y_2) \geq d(y,y_2) - d(y,y') \geq d - t$$

$d(y,y_2) \geq d(C) = d \geq 2t+1$

$d(y,y') \leq t \Rightarrow -d(y,y') \geq -t$

$$\Rightarrow d(y',y_2) \geq (2t+1) - t = t+1$$

$\therefore d(y,y') \leq t$ and $d(y_2,y') \geq t+1$ for all other codewords y_2

$\Rightarrow y'$ is closer to y than to y_2

$\Rightarrow y$ is a unique codeword satisfying $d(y,y') \leq t$.

Exercise 10.20: Let H be a parity check matrix such that any $d-1$ columns are linearly independent, but there exists a set of d linearly dependent columns. Show that the code defined by H has distance d.

Proof.

$$d(C) = \min_{\substack{x,y \in C \\ x \neq y}} d(x,y) = \min_{x \in C, x \neq 0} wt(x)$$

Part 1: Assume $d(C) = d$,

$\exists$ some codeword c with weight d

$\therefore Hc = 0$

$A\vec{v} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ & & & \\ a_{m1} & a_{m2} & & a_{mn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} a_{11}v_1 + a_{12}v_2 + \cdots + a_{1n}v_n \\ a_{21}v_1 + a_{22}v_2 + \cdots + a_{2n}v_n \\ \vdots \\ a_{m1}v_1 + a_{m2}v_2 + \cdots + a_{mn}v_n \end{bmatrix} = v_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + v_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots + v_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}$

$A\vec{v}$ is a linear combination of columns of A $\{ = v_1\vec{a}_1 + v_2\vec{a}_2 + \cdots + v_n\vec{a}_n$

$$Hc = \sum_{\substack{j \in 1 \text{ positions} \\ \text{of } c}} c_j \cdot H_j = \sum_{\substack{j \in 1 \text{ positions} \\ \text{of } c}} H_j = 0$$

$\Rightarrow$ d columns $\{H_j\}$ of H corresponding to the non-zero elements of c must be linearly dependent.

If $\exists$ m linearly dependent columns s.t $0 < m < d$ then $\exists$ some codeword x s.t $Hx = 0$ with $d(x) = m < d$

But this contradicts $d(C) = d$

$\Rightarrow$ $d-1$ columns are linearly independent.

Part 2: Assume $\exists$ d LD columns, but m s.t $0 < m < d$ columns are linearly independent.

Then, the solutions of $Hx = 0$ have a minimum weight d, and the solutions are codewords. $\Rightarrow d(C) = d$

Exercise 10.21: (Singleton bound) Show that an $[n, k, d]$ code must satisfy $n - k \geq d - 1$.

Proof. $H \in \mathbb{Z}_2^{(n-k)\times n}$

$(n-k)$ rows in $H \Rightarrow$ any $(n-k+1)$ columns of H are LD

$\therefore \quad d \leq n-k+1 \Rightarrow n-k \geq d-1$.

A good illustrative class of linear error-correcting codes are the Hamming codes. Suppose $r \geq 2$ is an integer and let H be the matrix whose columns are all $2^r - 1$ bit strings of length r which are not identically 0. This parity check matrix defines a $[2^r - 1, 2^r - r - 1]$ linear code known as a *Hamming code*. An especially important example for quantum error-correction is the case $r = 3$, which is a $[7, 4]$ code having parity check matrix:

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{10.62}$$

Any two columns of H are different, and therefore linearly independent; the first three columns are linearly dependent, so by Exercise 10.20 the distance of the code is 3. It follows that this code is able to correct an error on any single bit. Indeed, the error-correction method is very simple. Suppose an error occurs on the jth bit. Inspection of (10.62) reveals that the syndrome He_j is just a binary representation for j, telling us which bit to flip to correct the error.

Exercise 10.22: Show that all Hamming codes have distance 3, and thus can correct an error on a single bit. The Hamming codes are therefore $[2^r - 1, 2^r - r - 1, 3]$ codes.

Proof.

> $r \geq 2$ is an integer and let H be the matrix whose columns are all $2^r - 1$ bit strings of length r which are not identically 0. This parity check matrix defines a $[2^r - 1, 2^r - r - 1]$ linear code known as a *Hamming code*.

Each column differ in at least 1 bit $\Rightarrow H_i + H_j \neq 0$

No 2 columns in H are LD

Columns of H are binary representations of integers upto $2^r - 1$ excluding 0.

$\therefore H^{(1)} + H^{(2)} + H^{(3)} = 001 + 010 + 011 = 000 = 0$

$\Rightarrow \exists$ 3 LD columns in H

Ex 10.20 $\Rightarrow$ $d(C) = 3$.

$d = 2t+1 = 3 \Rightarrow t = 1$ } Hamming code can correct an error on a single bit.

We conclude our survey of classical error-correction by explaining an important construction for codes known as the *dual* construction. Suppose C is an $[n, k]$ code with generator matrix G and parity check matrix H. Then we can define another code, the *dual* of C, denoted $C^{\perp}$, to be the code with generator matrix H^T and parity check matrix G^T. Equivalently, the dual of C consists of all codewords y such that y is orthogonal to all the codewords in C.

* $C = \ker(H) = \text{Range}(G)$

$C^{\perp} = \ker(G^T) = \text{Range}(H^T) = \text{rowspace}(H)$

$\ker(H) \perp \text{rowspace}(H) \implies C \perp C^{\perp}$

$\dim(C) = k \implies \dim(C^{\perp}) = n - k$

* $C^{\perp} = \{ z \mid \langle x, z \rangle = 0 \ \forall \ x \in C \}$

* The dual of an $[n, k, d]$ code is an $[n, n-k, d]$ code.

(self orthogonal) (self dual)

- A code is said to be *weakly self-dual* if $C \subseteq C^{\perp}$, and *strictly self-dual* if $C = C^{\perp}$.

ie.,

$C \subseteq C^{\perp}$ iff C is self orthogonal, ie., every pair of codewords in C is orthogonal to each other.

$C = C^{\perp}$ iff C is self-dual ie., C is equal to its dual code.

Ex 10.18 $\Rightarrow$ $HG = HH^T = H\left[h_1^T \; h_2^T \cdots h_{n-k}^T\right]$ $H_{(n-k)\times n}$

$= \left[Hh_1^T \; Hh_2^T \cdots Hh_{n-k}^T\right] = 0$

$= \begin{bmatrix} h_1 \cdot h_1 & h_1 \cdot h_2 & \cdots \\ h_2 \cdot h_1 & h_2 \cdot h_2 & \cdots \\ & & \end{bmatrix} = 0$

$\therefore$ Every row of H is orthogonal to each row of H

$\therefore$ Every row x of H satisfies the equation $Hx^T = 0$

$\Rightarrow$ rowspace(H) is contained in the kernel(H).

$\dim(C) = \dim(C^{\perp}) \Rightarrow \dim(\text{rowspace}(H)) = \dim(\ker(H))$

$\dim(C) = n = n - k = \dim(C^{\perp}) \Rightarrow k = n - k = n/2$

if a code is self-dual, then its length n must be even, and its dimension must be $n/2$.

Ex:- Consider the parity check matrix H of the code C:

$$H = \left[\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right]$$

Compute $C\left(\ker(H)\right)$: $Hx = 0 \Rightarrow$ $x_1 + x_3 = 0 \mod 2$

$x_2 + x_4 = 0 \mod 2$

$\left[-x = x \mod 2\right]$

$x_3 = x_1$ & $x_4 = x_2$

$$x = x_1\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + x_2\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \text{where} \quad x_1, x_2 \in \{0,1\}.$$

$$\therefore \quad C = \{(0,0,0,0), (1,0,1,0), (0,1,0,1), (1,1,1,1)\}$$

Compute $C^{\perp}\left(\text{rowspace}(H)\right)$: All combinations of rows.

$$\therefore \quad \underline{C = C^{\perp}}.$$

$c_i \cdot c_j = 0$ for all $c_i, c_j \in C$

Calderbank-Shor-Steane (CSS) codes

Suppose C_1 and C_2 are $[n, k_1]$ and $[n, k_2]$ classical linear codes s.t $C_2 \subset C_1$, and C_1 & $C_2^{\perp}$ both correct t errors.

A coset of C_2 in C_1 is defined as:

$$x + C_2 = \{x + y \mid y \in C_2\} \qquad \text{where } |x + C_2| = |C_2|$$

where $x \in C_1$ is any codeword in the code C_1, and the addition is bitwise modulo 2.

The quantum state corresponding to the coset $x + C_2$ is defined as:

$$|x + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle$$

which is the uniform superposition of all codewords in the coset $x + C_2$.

Note: The Hilbert space is the space of all quantum states that can be represented by bitstrings of length n. i.e., $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$, the 2^n dimensional space for n-qubit systems, with basis states $\{|z\rangle \mid z \in \mathbb{F}_2^n\}$ where z is a binary string of length n.

$$\langle z | z' \rangle = \begin{cases} 1 & \text{if } z = z' \\ 0 & \text{if } z \neq z' \end{cases}$$

If $x' \in C_1$ is another element from the same coset, ie., $x' = x + c_2$ for some $c_2 \in C_2$ (or) $x - x' \in C_2$ then

$$|x + C_2\rangle = |x' + C_2\rangle$$

$\Rightarrow$ The quantum state $|x+C_2\rangle$ depends only on the coset $x+C_2$ in C_1/C_2 which x is in, not on the specific representative x chosen.

- C_2 partitions C_1 into disjoint cosets.

If x, x' belongs to different cosets of C_2, then the cosets $x+C_2$ and $x'+C_2$ are distinct, and the corresponding quantum states are:

$$|x + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x+y\rangle$$

$$|x' + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y' \in C_2} |x'+y'\rangle$$

$$\langle x + C_2 | x' + C_2\rangle = \frac{1}{|C_2|} \sum_{y, y' \in C_2} \langle x+y | x'+y'\rangle$$

If $x + C_2 \neq x' + C_2$ (the cosets are distinct),

no $y, y' \in C_2$ satisfy $x+y = x'+y'$ because cosets are disjoint.

$\therefore \quad \langle x+y | x'+y'\rangle = 0 \quad \forall \quad y, y' \in C_2$

$\therefore \quad \langle x + C_2 | x' + C_2\rangle = 0$ for distinct cosets.

$\therefore |x+C_1\rangle$ and $|x+C_2\rangle$ are orthonormal states.

The # of cosets of C_2 in $C_1 = (C_1 : C_2) = \frac{|C_1|}{|C_2|}$

$$= \frac{2^{k_1}}{2^{k_2}} = 2^{k_1-k_2}$$

$\therefore$ The quantum code $CSS(C_1, C_2)$ is defined to be vector space spanned by the states $|x+C_2\rangle$ for all $x \in C_1$, with dimension $|C_1|/|C_2| = 2^{k_1-k_2}$.

$$CSS(C_1, C_2) = \text{span}\{|x+C_2\rangle : x \in C_1\}$$

The quantum code encodes $k = k_1 - k_2$ logical qubits into n physical qubits, and the $CSS(C_1, C_2)$ code is an $[n, k_1-k_2]$ quantum code.

Suppose the bit flip errors are described by an n-bit vector e_1 with 1's where bit flip occurred, and 0's elsewhere, & phase flip errors are described by an n-bit vector e_2 with 1's where phase flip occurred, and 0's elsewhere.

$$X^{e_1}|z\rangle = |z+e_1\rangle \quad \text{and}$$

for each qubit i,

if $e_2[i]=1$ and $z[i]=1$, a phase flip occurs, introducing a factor of -1

if $z[i]=0$, no phase flip occurs

$\Rightarrow$ The overall phase factor is determined by counting the # of qubits where $z[i]=1$ and $e_2[i]=1$.

ie: the dot product modulo 2

$$z \cdot e_2 = \sum_{i=1}^{n} z[i].e[i] \mod 2$$

$$\therefore \quad Z^{e_2}|z\rangle = (-1)^{z \cdot e_2}|z\rangle$$

$\therefore$ If $|x+C_2\rangle$ was the original state then the corrupted state is:

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y)\cdot e_2} |x+y+e_1\rangle$$

To detect where bit flips occurred it is convenient to introduce an ancilla containing sufficient qubits to store the syndrome for the code C_1, and initially in the all zero state $|0\rangle$. We use reversible computation to apply the parity matrix H_1 for the code C_1, taking

$$|x+y+e_1\rangle|0\rangle \longrightarrow |x+y+e_1\rangle|H_1(x+y+e_1)\rangle = |x+y+e_1\rangle|H_1 e_1\rangle$$

since $H_1((x+y)+e_1) = H_1(x+y) + H_1 e_1 = 0 + H_1 e_1 = H_1 e_1$ given $x+y \in C_1$.

Exercise 10.26: Suppose H is a parity check matrix. Explain how to compute the transformation $|x\rangle|0\rangle \rightarrow |x\rangle|Hx\rangle$ using a circuit composed entirely of controlled-NOTs.

Ans: $U_{CNOT}|a\rangle|b\rangle = |a\rangle|a \oplus b\rangle \Rightarrow U_{CNOT}|a\rangle|0\rangle = |a\rangle$

i) Start with $|a\rangle$ on n qubits and $m = n-k$ qubits initialized to $|0\rangle$ (ancilla).

ii) For each row i of H,

for each column j such that $H_{ij} = 1$,

apply CNOT gate with x_j as the control qubit and i^{th} ancilla qubit as the target.

The effect of the operation is to produce the state:

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y)\cdot e_2} |x+y+e_1\rangle |H_1 e_1\rangle$$

Error-detection for the bit flip errors is completed by measuring the ancilla to obtain the result $H_1 e_1$ and discarding the ancilla, giving the state

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y)\cdot e_2} |x + y + e_1\rangle. \tag{10.67}$$

Knowing the error syndrome $H_1 e_1$ we can infer the error e_1 since C_1 can correct up to t errors, which completes the error-detection. Recovery is performed simply by applying NOT gates to the qubits at whichever positions in the error e_1 a bit flip occurred, removing all the bit flip errors and giving the state

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y)\cdot e_2} |x + y\rangle. \tag{10.68}$$

For a single qubit, $\boxed{H|x\rangle = \sum_z \frac{(-1)^{xz}|z\rangle}{\sqrt{2}}}$

$$H^{\otimes n}|x_1,\cdots,x_n\rangle = \sum_{z_1} \frac{(-1)^{x_1 z_1}|z_1\rangle}{\sqrt{2}} \otimes \sum_{z_2} \frac{(-1)^{x_2 z_2}|z_2\rangle}{\sqrt{2}} \otimes \cdots \otimes \sum_{z_n} \frac{(-1)^{x_n z_n}|z_n\rangle}{\sqrt{2}}$$

$$= \sum_{z_1 z_2 \cdots z_n} \frac{(-1)^{x_1 z_1 + x_2 z_2 + \cdots + x_n z_n}|z_1 z_2 \cdots z_n\rangle}{\sqrt{2^n}}$$

$$= \sum_z \frac{(-1)^{x\cdot z}|z\rangle}{\sqrt{2^n}}$$

$\therefore$

$$H^{\otimes n}|x+y\rangle = \sum_z \frac{1}{\sqrt{2^n}} (-1)^{(x+y)\cdot z}|z\rangle$$

To detect phase flip errors, we apply Hadamard gates to each qubit, taking the state to

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y)\cdot e_2} |x+y\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{|C_2|2^n}} \sum_z \sum_{y \in C_2} (-1)^{(x+y)\cdot(e_2+z)} |z\rangle$$

where the sum is over all possible values for n bit z.

Setting $z' \equiv z + e_2$,

$$\frac{1}{\sqrt{|C_2|2^n}} \sum_{z'} \sum_{y \in C_2} (-1)^{(x+y)\cdot z'} |z' + e_2\rangle$$

Exercise 10.25: Let C be a linear code. Show that if $x \in C^\perp$ then $\sum_{y \in C}(-1)^{x\cdot y} = |C|$, while if $x \notin C^\perp$ then $\sum_{y \in C}(-1)^{x\cdot y} = 0$.

Proof If $x \in C^\perp$ then,

$x \cdot y = 0 \quad \forall \; y \in C$

$$\therefore \sum_{y \in C} (-1)^{x\cdot y} = \sum_{y \in C} 1 = |C|$$

$C^\perp = \{x \mid x \cdot y = 0 \; \forall \; y \in C\}$

If $x \notin C^\perp$ then,

$\exists$ some $y^* \in C$ s.t $x \cdot y^* = 1$

C is a subspace $\Rightarrow$ If $y \in C$, then $y + y^* \in C$ for any codeword $y^* \in C$.

$$2\sum_{y \in C} (-1)^{x\cdot y} = \sum_{y \in C} (-1)^{x\cdot y} + \sum_{y \in C} (-1)^{x\cdot(y+y^*)}$$

$$= \sum_{y \in C} \left[(-1)^{x\cdot y} - (-1)^{x\cdot y} \right] = 0$$

$y + y^* = y'$

$y_1 + y^* = y'$

$y - y_1 = y' - y' = 0$

$$\therefore \sum_{y \in C_2} (-1)^{y \cdot z'} = \begin{cases} |C_2| & \text{if } z' \in C_2^{\perp} \\ 0 & \text{if } z' \notin C_2^{\perp} \end{cases}$$

$$\frac{1}{\sqrt{|C_2|2^n}} \sum_{z'} \sum_{y \in C_2} (-1)^{(x+y)\cdot z'} |z' + e_2\rangle = \frac{1}{\sqrt{|C_2|2^n}} \sum_{z'} \left(\sum_{y \in C_2} (-1)^{(x+y)\cdot z'} \right) |z' + e_2\rangle$$

$$= \frac{1}{\sqrt{2^n/|C_2|}} \sum_{z' \in C_2^{\perp}} (-1)^{x \cdot z'} |z' + e_2\rangle$$

which looks just like a bit flip error described by the vector e_2! As for the error-detection for bit flips we introduce an ancilla and reversibly apply the parity check matrix H_2 for $C_2^{\perp}$ to obtain $H_2 e_2$, and correct the 'bit flip error' e_2,

(10.72)

Since $z' \in C_2^{\perp}$, $H_2 |z'\rangle = 0$

$$\therefore H_2(z' + e_2) = H_2 z' + H_2 e_2 = H_2 e_2$$

$$\therefore |z' + e_2\rangle |0\rangle \rightarrow |z' + e_2\rangle |H(z' + e_2)\rangle = |z' + e_2\rangle |H e_2\rangle$$

The effect of this operation is the state

$$\frac{1}{\sqrt{2^n/|C_2|}} \sum_{z' \in C_2^{\perp}} (-1)^{x \cdot z'} |z' + e_2\rangle |H_2 e_2\rangle$$

Measuring the ancilla to obtain the error syndrome $H_2 e_2$, thereby can infer the error e_2 since C_2 can correct upto t errors. Recovery is performed by applying NOT gates to the qubits at whichever positions in the error e_2 a bit flip occurred, giving the state:

$$\frac{1}{\sqrt{2^n/|C_2|}} \sum_{z' \in C_2^\perp} (-1)^{x \cdot z'} |z'\rangle$$

Now,

$$\frac{1}{\sqrt{2^n/|C_2|}} \sum_{z' \in C_2^\perp} (-1)^{x \cdot z'} |z' + e_2\rangle = \frac{1}{\sqrt{|C_2|2^n}} \sum_{z'} \sum_{y \in C_2} (-1)^{(x+y) \cdot z'} |z' + e_2\rangle$$

$$= \frac{1}{\sqrt{|C_2|2^n}} \sum_{z} \sum_{y \in C_2} (-1)^{(x+y) \cdot (e_2 + z)} |z\rangle$$

$$= H^{\otimes n} \left(\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y) \cdot e_2} |x + y\rangle \right)$$

For $e_2 = 0$,

$$\frac{1}{\sqrt{2^n/|C_2|}} \sum_{z' \in C_2^\perp} (-1)^{x \cdot z'} |z'\rangle = H^{\otimes n} \left(\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle \right)$$

The error correction is completed by again applying Hadamard gates to each qubit, in eq^n. 10.72.

$$\therefore \quad H^{\otimes n} \left(\frac{1}{\sqrt{2^n/|C_2|}} \sum_{z' \in C_2^\perp} (-1)^{x \cdot z'} |z'\rangle \right) = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle$$

$$= |x + C_2\rangle$$

which is the original encoded state!

Summarizing, suppose C_1 and C_2 are $[n, k_1]$ and $[n, k_2]$ classical linear codes, respectively, such that $C_2 \subset C_1$, and both C_1 and $C_2^{\perp}$ can correct errors on up to t bits. Then CSS(C_1, C_2) is an $[n, k_1 - k_2]$ quantum error-correcting code which can correct arbitrary errors on up to t qubits. Furthermore, the error-detection and correction steps require only the application of Hadamard and controlled-NOT gates, in each case a number linear in the size of the code. Encoding and decoding can also be performed using a number of gates linear in the size of the code

The Steane code

The Stabilizer Formalism.

$X|0\rangle = |1\rangle$
$X|1\rangle = |0\rangle$

$Z|0\rangle = |0\rangle$
$Z|1\rangle = -|1\rangle$

EPR state of two qubits

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

$X_1X_2|\psi\rangle = |\psi\rangle$ and $Z_1Z_2|\psi\rangle = |\psi\rangle$

$\Longrightarrow$ the state $|\psi\rangle$ is stabilized by the operators X_1X_2 and Z_1Z_2.

- The state $|\psi\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ is the unique quantum state (up to a global phase) which is stabilized by the operators X_1X_2 and Z_1Z_2.

The basic idea of the stabilizer formalism is that many quantum states can be more easily described by working with the operators that stabilize them than by working explicitly with the state itself.

It turns out that many quantum codes (including CSS codes and the Shor code) can be much more compactly described using stabilizers than in the state vector description. Even more importantly, errors on the qubits and operations such as the Hadamard gate, phase gate, and even the controlled-NOT gate and measurements in the computational basis are all easily described using the stabilizer formalism!

Pauli group G_n on n qubits. For a single qubit, the Pauli group is defined to consist of all the Pauli matrices, together with multiplicative factors $\pm 1, \pm i$:

$$G_1 \equiv \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}. \quad (10.81)$$

This set of matrices forms a group under the operation of matrix multiplication.

The general Pauli group on n qubits is defined to consist of all n-fold tensor products of Pauli matrices, and again we allow multiplicative factors $\pm 1, \pm i$.

A group $(G, \cdot)$ is a non-empty set G with a binary group multiplication operation '$\cdot$', with the following properties :

(i) Closure : $g_1 \cdot g_2 \in G$ for all $g_1, g_2 \in G$

(ii) Associativity : $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$ for all $g_1, g_2, g_3 \in G$

(iii) Identity : there exists $e \in G$ such that $\forall g \in G$, $g \cdot e = e \cdot g = g$

(iv) Inverses : For all $g \in G$, there exists $g^{-1} \in G$ such that $g g^{-1} = e$ and $g^{-1} \cdot g = e$

- Let $(G, *)$ be a group. A non-empty subset H of G is called a subgroup of G if, ie., $H \leq G$

  (i) $a * b \in H \quad \forall \; a, b \in H$
  i.e., $*$ is a binary operation on H

  (ii) $(H, *)$ is itself a group.

Given $H \subseteq G$,
$H \leq G \iff a, b \in H \Rightarrow ab^{-1} \in H$

Ex:- $(\mathbb{Z}, +)$ is a subgroup of $(\mathbb{Q}, +)$, $(\mathbb{R}, +)$ and $(\mathbb{C}, +)$.

Suppose S is a subgroup of G_n and define V_S to be the set of n qubit states which are fixed by every element of S. V_S is the *vector space stabilized by* S, and S is said to be the *stabilizer* of the space V_S, since every element of V_S is stable under the action of elements in S.

At Classmate, we are committed to providing high quality stationery products that are a result of a deep understanding of our consumers, thoughtful ideation, innovative designs and superior craftsmanship, that have, in turn, helped us become one of India's leading stationery brands!

classmate

Our Classmate range of products include: **NOTEBOOKS,** Writing Instruments - **PENS** (ball, gel and roller), **PENCILS** (mechanical), **MATHEMATICAL DRAWING INSTRUMENTS, ERASERS, SHARPENERS** and **ART STATIONERY** (wax crayons, colour pencils, sketch pens and oil pastels).

ITC's Primary Education Programme is designed to provide children from weaker sections access to learning with special focus on quality and retention. The Read India Plus Programme, run in partnership with Pratham, helps improve the quality of learning of around 36,000 children every year. In addition, ITC has helped set up supplementary learning centers and strengthened infrastructure in government schools. So far, over 4,00,000 children have benefitted.

ITC Enduring Value

Let's put India first

Customize your notebook covers at **www.classmateshop.com**

WHITE & BRIGHT PAPER | OZONE TREATED | GREEN PAPER

Classmate uses eco-friendly and elemental chlorine free paper

Scan with your smartphone.
Visit us at
www.classmateshop.com

FEEDBACK ? SUGGESTIONS ?

Quality Manager, ITC Ltd.- ESPB,
ITC Centre, 5th Floor,
760, Anna Salai, Chennai. 600 002.
classmate@itc.in 18004253242
(Toll-Free from MTNL/BSNL lines)

A quality product marketed by

ITC Enduring Value

Exercise Book

172 Pages
(Total Pages Include Index & Printed Information)

Size : 24 x 18 cm

MRP Rs. 48.00
Inclusive of all taxes

Batch: K/AE/FT

02000222

8 902519 002228

Type of Ruling :

Unruled

TYNF