If $r \ll 2^n$: many values of k & almost periodic

$\Rightarrow |a_\ell|^2$ peaked around ℓ s.th. $\frac{r\ell}{2^n} \approx$ integer

Explicit analysis of a_ℓ shows: w.h.p., we obtain ℓ s.th.

↳ "with high probability"

$$\frac{\ell}{2^n} \approx \frac{s}{r}$$

If $r \ll 2^n$, this can be used to determine $\frac{s}{r}$ w.h.p.

If s, r are coprime (i.e., $\gcd(s,r) = 1$) – this happens with large enough prob., related to density of primes – we can infer r!

$\Rightarrow$ Quantum Algorithm for period finding!

c) Application: factoring

One use of of period finding: factoring.

Given N (not prime) $\rightarrow$ find non-trivial r: $r \mid N$

↑

"divides"

Algorithm:

① Select random a, $2 \leq a < N$.

If $\gcd(a,N) \geq 1 \Longrightarrow$ done!

↳ eff. computable! (Euclid's algorithm)

So assume $\gcd(a,N) = 1$.

② Let r be the smallest r s.th. $a^r \bmod N = 1$.

(Existence: (!) $\exists x,y: a^x \equiv a^y \bmod N \Rightarrow a^x(1-a^{y-x}) \equiv 0 \bmod N$

$\Rightarrow N \mid a^x(1-a^{y-x}) \overset{\gcd(a,N)=1}{\Longrightarrow} N \mid 1-a^{y-x} \Rightarrow a^{y-x} \equiv 1 \bmod N$)

r is the period of $f_{N,a}(x) = a^x \bmod N$

$f_{N,a}(x)$ can be computed efficiently:

with $x = x_{m-1}2^{m-1} + x_{m-2}2^{m-2} + \dots$,

$$a^x \bmod N = \underbrace{\left(a^{(2^{m-1})}\right)^{x_{m-1}} \cdot \left(a^{(2^{m-2})}\right)^{x_{m-2}} \cdot \dots}_{\text{eff. computable by } a \mapsto a^2 \mapsto a^4 \mapsto a^8 \mapsto \dots} \bmod N,$$

$\Rightarrow r$ can be found eff. w/ a quantum computer!

(3) Assume r even:

$$a^r \bmod N = 1 \iff N \mid (a^r - 1) \iff N \mid (a^{r/2} - 1)(a^{r/2} + 1)$$

and $N \nmid (a^{r/2} - 1)$ (otherwise, $a^{r/2} \bmod N = 1$ ↯)

$\Rightarrow$ either $N \mid a^{r/2} + 1$

or N has non-triv. common factors with both $a^{r/2} \pm 1$

$\Rightarrow 1 \neq \gcd(N, a^{r/2} \pm 1) \mid N$

$\Rightarrow$ found non-triv. factor of N!

$\Rightarrow$ Algorithm successful as long as
(i) r even & (ii) $N \nmid (a^{r/2} + 1)$

Can be shown to happen w/ $p \geq 1/2$ for random choice of a (unless $N = p^k$, p prime → can be checked by taking logs)

$\Rightarrow$ <u>efficient quantum algorithm for factoring</u>

<u>"Shor's algorithm"</u>

V. Quantum Error Correction

V.1. Introduction

- Coupling to environment reduces errors
- Classical computers: "macroscopic" → errors unlikely
- Q. computers: – need qubits = "single" quantum systems
  → fragile!
  – need coupling to "env." to realize gates!

So... can we protect quantum information from noise?

Classical error correction:

copy information, e.g. encode 1 bit in 3:

$$0 \longmapsto \hat{0} = 000$$
$$1 \longmapsto \hat{1} = 111$$

Bit flip w/ some (small) probability $p \Rightarrow$ typ. 0 or 1 bits flipped

Correction: majority vote, i.e.

$$000, 001, 010, 100 \longmapsto 000$$
$$111, 110, 101, 011 \longmapsto 111$$

$p_{error} = \text{prob}(\geq 2 \text{ flips}) = p^3 + 3p^2(1-p) \leq 3p^2 < p$ for $p < 1/3$!

⇒ effective error prob. decreased.

Can be improved by
- → using more bits
- → using smarter codes (encoding k bits)
- → nesting ("concatenating") codes

Quantum error correction:

Several potential problems:

- cannot copy qubits (and even if: how do we compare them?)
- different types of errors, e.g. X (bit flip) or Z (phase flip)
- errors can be continuous
- measuring qubits destroys q. info!

a) The 3-qubit bit flip code

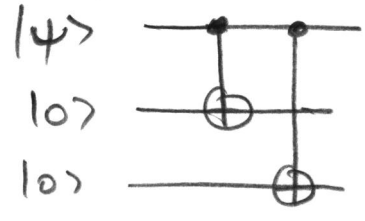
Copy qubits in fixed steps:

$$|0\rangle \longmapsto |\hat{0}\rangle = |000\rangle$$

$$|1\rangle \longmapsto |\hat{1}\rangle = |111\rangle$$

i.e.: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \xrightarrow{\text{encoding}} |\hat{\psi}\rangle = \alpha|000\rangle + \beta|111\rangle$

Encoding circuit:



$\Rightarrow |\hat{\psi}\rangle = \alpha|000\rangle + \beta|111\rangle$

Consider bit flip error $|\hat{\psi}\rangle \mapsto X_i|\hat{\psi}\rangle$ on qubit i;

Can we correct for one bit flip error on unknown qubit?

Problem: Meas. all qubits destroys q. info!

$\Rightarrow$ Need meas. which only returns info about location of error (indep. of encoded state $|\psi\rangle$).

Define "syndrome measurement" with projectors:

"no flip": $P_0 = |000\rangle\langle 000| + |111\rangle\langle 111|$

"1st qubit flipped": $P_1 = |100\rangle\langle 100| + |011\rangle\langle 011|$

"2nd qubit flipped": $P_2 = |010\rangle\langle 010| + |101\rangle\langle 101|$

"3rd qubit flipped": $P_3 = |001\rangle\langle 001| + |110\rangle\langle 110|$

Measuring $\{P_\alpha\}$ reveals only 2 bits of information

$\Rightarrow$ one qubit untouched,

By inspection: info. obtained $\equiv$ location of flip, e.g.

$$\alpha|000\rangle + \beta|111\rangle \xrightarrow[\text{qubit 1}]{\text{flip of}} \alpha|100\rangle + \beta|011\rangle$$

$\Rightarrow$ meas. returns P_1, part meas. state

$$\alpha|100\rangle + \beta|011\rangle \xrightarrow[\text{qubit 1}]{\text{recovery: flip}} \alpha|000\rangle + \beta|111\rangle$$

Works for any single or no flipped qubit, & all states $|\psi\rangle$.

(Linearity: also works for parts of large entangled state.)

What about continuous errors, such as

$$|\hat{\psi}\rangle \longmapsto e^{i\vartheta X_i}|\hat{\psi}\rangle = (\cos\vartheta\, \mathbb{1} + i \sin\vartheta\, X_i)|\hat{\psi}\rangle \; ?$$

$$|\hat{\psi}\rangle = \alpha|000\rangle + \beta|111\rangle \xrightarrow[\text{qubit 1}]{\text{error, eg}} \cos\vartheta \underbrace{(\alpha|000\rangle + \beta|111\rangle)}_{\text{Syndrome } P_0} + i \sin\vartheta \underbrace{(\alpha|100\rangle + \beta|011\rangle)}_{\text{Syndrome } P_1}$$

Syndrome meas. collapses state to:

$p = \cos^2\vartheta$: $P_0 \Rightarrow \alpha|000\rangle + \beta|111\rangle$, no correction ✓

$p = \sin^2\vartheta$: $P_1 \Rightarrow \alpha|100\rangle + \beta|011\rangle$, flip bit 1 ✓

Meas. of error syndrome P_i collapses error onto "digital" error — no error or bit flip $\Rightarrow$ sufficient to study discrete errors!

What about Z errors?

$$\alpha|000\rangle + \beta|111\rangle \xrightarrow[\text{on some qubit}]{\text{Z error}} \alpha|000\rangle - \beta|111\rangle$$

This is still in code space (i.e., space of valid $|\psi\rangle$)

$\Rightarrow$ error not detectable, but $|\psi\rangle$ has changed!

$\Rightarrow$ 3-qubit bit flip code cannot protect against "phase flip" Z.

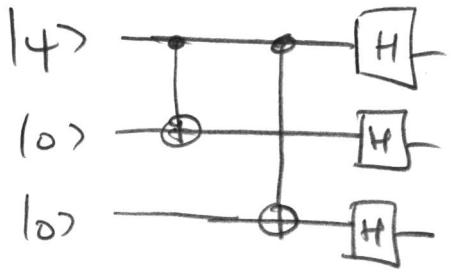
In fact: Phase flip Z_i acts as logical Z on the encoded qubit.

6) 3-qubit phase flip code

We have $Z|+\rangle = |-\rangle$, $Z|-\rangle = |+\rangle$

Z error $\hat{=}$ bit flip error in $|\pm\rangle$-basis.

Encoding $|\hat{0}\rangle = |+++\rangle$, $|\hat{1}\rangle = |---\rangle$ will protect against Z errors!



Syndrome meas. $\tilde{P}_\alpha := H^{\otimes 3} P_\alpha H^{\otimes 3}$, recovery $HX_iH = Z_i$.

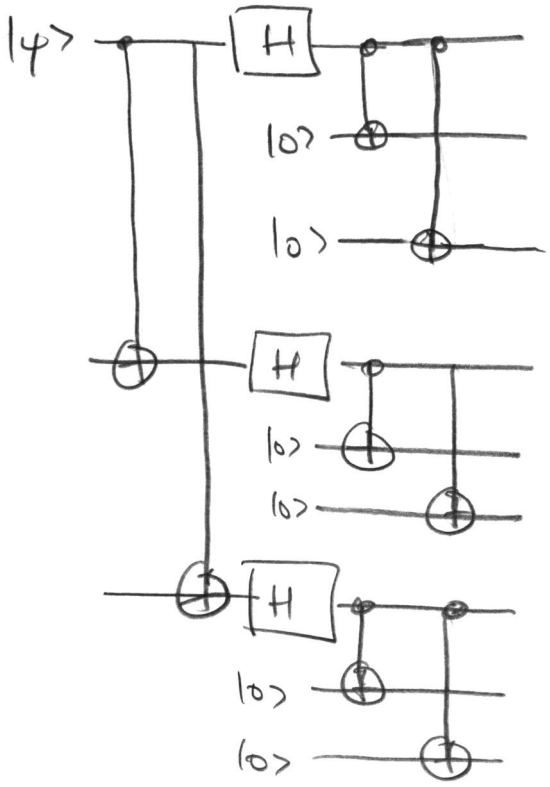
Problem: Now, no protection against bit-flip errors.
(and X_i acts as X on logical qubit)

V.2. The 9-qubit Shor code

Solution: Concatenate (= nest) 3-qubit bit flip with 3-qubit phase flip code!

$$|0\rangle \longmapsto |+\rangle|+\rangle|+\rangle \longmapsto \frac{(|000\rangle+|111\rangle)(|000\rangle+|111\rangle)(|000\rangle+|111\rangle)}{2\sqrt{2}}$$

$$|1\rangle \longmapsto |-\rangle|-\rangle|-\rangle \longmapsto \frac{(|000\rangle-|111\rangle)(|000\rangle-|111\rangle)(|000\rangle-|111\rangle)}{2\sqrt{2}}$$



9-qubit Shor code

Can correct any single-qubit Pauli:

(i) X_i error is corrected at "inner" layer.

(ii) Z_i error ≡ logical error on "outer" qubit

⟹ $Z_{block(i)}$ error on "outer" code (phase-flip)

⟹ correctable!

(iii) $Y_i \propto X_i Z_i$: X_i & Z_i corrected independently.