Lecture 3 : QFT & Phase Estimation

Both Deutsch-Jozsa & Simon's algorithms made good use of the Hadamard transformation $H^{\otimes n}$ to generate a useful superposition of computational basis states (or the inverse transformation - also $H^{\otimes n}$ - to extract useful information from such a superposition).

Classically, the Fourier transform is another extremely useful (and efficiently implementable) transformation that renders many problems tractable.

The quantum Fourier transform (which is really just the classical discrete Fourier transform carried out in Hilbert space) is equally useful, and will ultimately let us solve a whole class of <u>useful</u> problems (famously including factoring) <u>exponentially</u> faster that the best known classical algorithms.

1. Quantum Fourier Transform

"Normal" FT on functions:

$$f(x) \longrightarrow F(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx\, e^{2\pi i k x} f(x)\, dx$$

Discrete FT on vectors (of length L):

$$\vec{f}_x \longrightarrow \vec{F}_k = \frac{1}{\sqrt{L}} \sum_{x=0}^{L-1} e^{2\pi i k x/L} \vec{f}_x$$

DFT on vectors of length 2^n:

$$\vec{f}_x \longrightarrow \vec{F}_k = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i k x/2^N} \vec{f}_x$$

Manifestly a linear transformation
i.e. $(\vec{f}+\vec{g})_x \longrightarrow (\vec{F}+\vec{G})_k$
$\Rightarrow$ suffices to know how it acts on a basis, e.g. elementary vectors $\vec{j}$ with a "1" in j'th entry, 0's elsewhere:

$$\vec{j}_x \longrightarrow \vec{E}_k = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i k x/2^n} \vec{j}_x$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i k x/2^n} \delta_{jx}$$

$$= \frac{1}{\sqrt{2^n}} e^{2\pi i k j/2^n}$$

or, writing equation for whole vector at once, instead of component-wise:

$$\vec{j} \longrightarrow FT(\vec{j}) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k_j / 2^n} \cdot \vec{k}$$

or, writing our vectors in Dirac notation,

$$|j\rangle \longrightarrow QFT|j\rangle = \frac{1}{\sqrt{2^N}} \sum_{k=0}^{2^N-1} e^{2\pi i j k / 2^n} |k\rangle$$

This is the QFT!

Exercise: Show the QFT is unitary. What about the FT on functions?

Efficient QFT implementation

Is there a poly(n)-sized circuit for the QFT?

$j, k = 0 \ldots 2^n \longrightarrow$ write them in binary:

$j = j_0 j_1 \cdots j_n$, $j_i \in \{0,1\}$ & similarly for k.

Then unitary we want on n qubits is:

$$QFT_{2^n}|j_0 j_1 \cdots j_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \omega_{2^n}^{jk} |k_0 k_1 \cdots k_n\rangle$$

where $\omega_{2^n} = e^{2\pi i / 2^n}$.

Examples:

$n = 1$: $$QFT_2 |j\rangle = \frac{1}{\sqrt{2}} \sum_{k=0}^{1} (-1)^{jk} |k\rangle$$

$$= \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^j |1\rangle \right)$$

$\rightarrow$ as a matrix $$QFT_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

This should look familiar: it's just our old friend, the single-qubit Hadamard!

$n = 2$:

$$QFT_4 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

Generally:

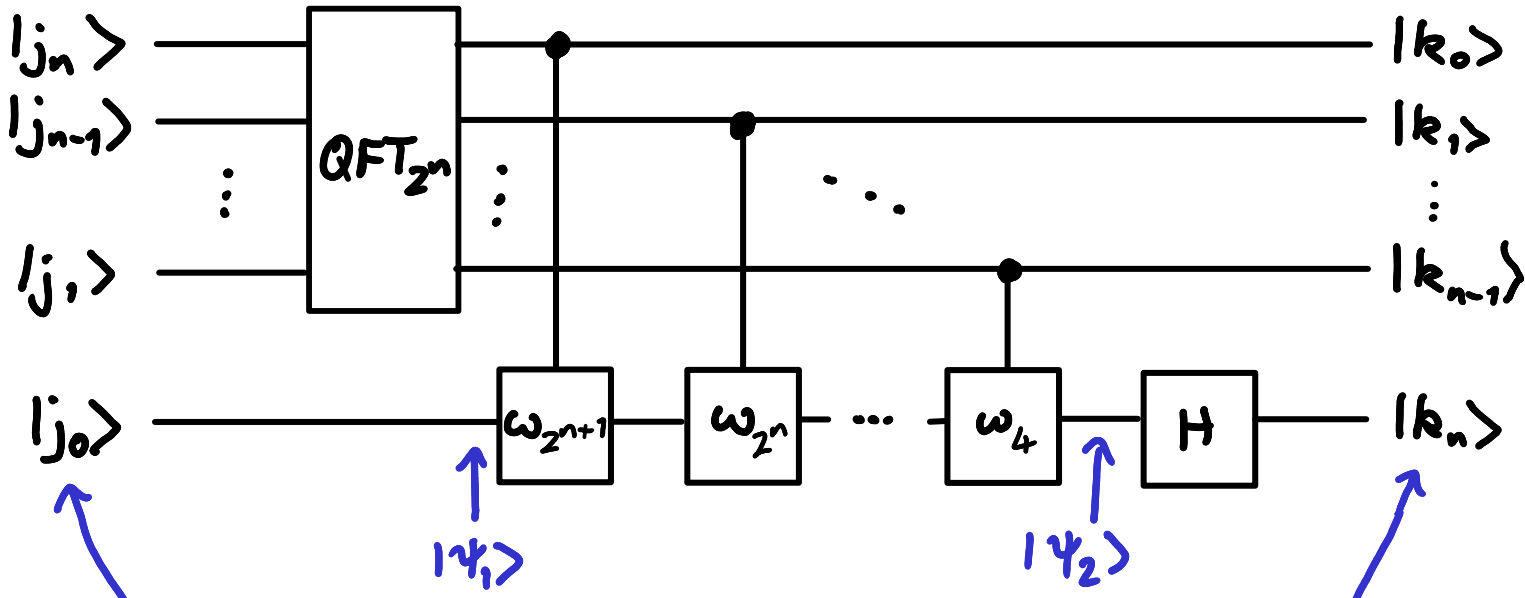
$$QFT_{2^n} = \frac{1}{\sqrt{2^n}} \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{2^n-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(2^n-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{2^n-1} & \omega^{2(2^n-1)} & \cdots & & \omega^{(2^n-1)^2} \end{pmatrix}$$

Construct circuit for QFT_{2^n} recursively:

Essentially this is the classical fast Fourier transform (FFT) algorithm carried out in Hilbert space.

QFT_{2^1}: base case

$QFT_{2^{n+1}}$: inductive case



Note: reverse order (can use SWAP gates to invert if needed)

Need to show this circuit gives $QFT_{2^{n+1}}$ assuming QFT_{2^n} works.

Let $j' = j_n j_{n-1} \cdots j_1 = \lfloor j/2 \rfloor$ (in binary)

$k' = k'_{n-1} k'_{n-2} \cdots k_0$

$\lfloor x \rfloor$ = x rounded down to nearest integer

$$|\psi_1\rangle = (QFT_{2^n} \; |j_n j_{n-1} \cdots j_1\rangle) \; |j_0\rangle$$

$$= (QFT_{2^n} \; |j'\rangle) \; |j_0\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{k'=0}^{2^n-1} \omega_{2^n}^{j'k'} \; |k'\rangle \; |j_0\rangle$$

$$|\psi_2\rangle = [\text{controlled phase shifts}] \; |\psi_1\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{k'=0}^{2^n-1} \omega_{2^n}^{j'k'} \; |k'\rangle \; \omega_{2^{n+1}}^{j_0 k'_0} \; \omega_{2^n}^{j_0 k'_1} \cdots \omega_4^{j_0 k'_{n-1}} \; |j_0\rangle$$

recall our QFT swap order of bits

$$= \frac{1}{\sqrt{2^n}} \sum_{k'=0}^{2^n-1} \omega_{2^{n+1}}^{2j'k' + j_0(k'_0 + 2k'_1 + \cdots + 2^{n-1} k'_{n-1})} \; |k'\rangle |j_0\rangle$$

using $\omega_{rN}^r = (e^{2\pi i/rN})^r = e^{2\pi i/N} = \omega_N$

$$= \frac{1}{\sqrt{2^n}} \sum_{k'=0}^{2^n-1} \omega_{2^{n+1}}^{jk'} \; |k'\rangle \; |j_0\rangle$$

$$QFT_{2^{n+1}} |j_n j_{n-1} \cdots j_0\rangle$$

$$= \mathbb{1}^{\otimes n} \otimes H \; |\psi_2\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{k'=0}^{2^n} \omega_{2^{n+1}}^{jk'} |k'\rangle \cdot \frac{1}{\sqrt{2}} \sum_{k_n=0}^{1} (-1)^{j_0 k_n} |k_n\rangle$$

$$= \frac{1}{\sqrt{2^{n+1}}} \sum_{k'} \sum_{k_n} \omega_{2^{n+1}}^{jk'} \omega_2^{j k_n} |k'\rangle |k_n\rangle$$

$$\omega_2^{jk_n} = (-1)^{jk_n} = (-1)^{(j_0 + 2j_1 + 2^2 j_2 \cdots) k_n} = (-1)^{jk_n}$$

$$= \frac{1}{\sqrt{2^{n+1}}} \sum_{k} \omega_{2^{n+1}}^{jk} |k\rangle \quad \text{as required.}$$

How many gates does circuit contain?

Let $g(n) = \#$ gates in QFT_{2^n}.

$$\begin{cases} g(1) = 1 & \text{Hadamard} \\ g(n+1) = g(n) + n + 1 & \leftarrow \text{final Hadamard} \end{cases}$$

($g(n)$: QFT_n; n: controlled-phase gates)

$$\Rightarrow g(n) = \sum_{j=1}^{n} j \leq n^2 \longrightarrow \text{efficient.}$$

Can achieve $O(n \log n)$ using fast-multiplication techniques, but constants are very large.

However, we've glossed over one issue: QFT circuit requires $\omega_{2^n} = e^{2\pi i/2^n}$ → exponentially small (in n) phase rotations.

Cannot construct these efficiently using standard gate set (cf. Solovay-Kitaev).

However, approximating exponentially small rotations by $\mathbb{1}$ gates (i.e. dropping them) only introduces exponentially small error in output state, which suffices for most applications, made rigorous in the following:

<u>Exercise</u>: Prove that state $|\tilde{\psi}\rangle$ produced by modified QFT circuit dropping phase rotations $2\pi i/2^n \leq \delta$ is close to state $|\psi\rangle$ produced by ideal QFT circuit:

$$\| |\tilde{\psi}\rangle - |\psi\rangle \|_1 = O(\delta \log n)$$

Hint: You may use without proof (though the proof is not difficult!) that

$$\| e^{-i\tilde{H}} - e^{-iH} \| \leq \| \tilde{H} - H \|.$$

2. Phase Estimation

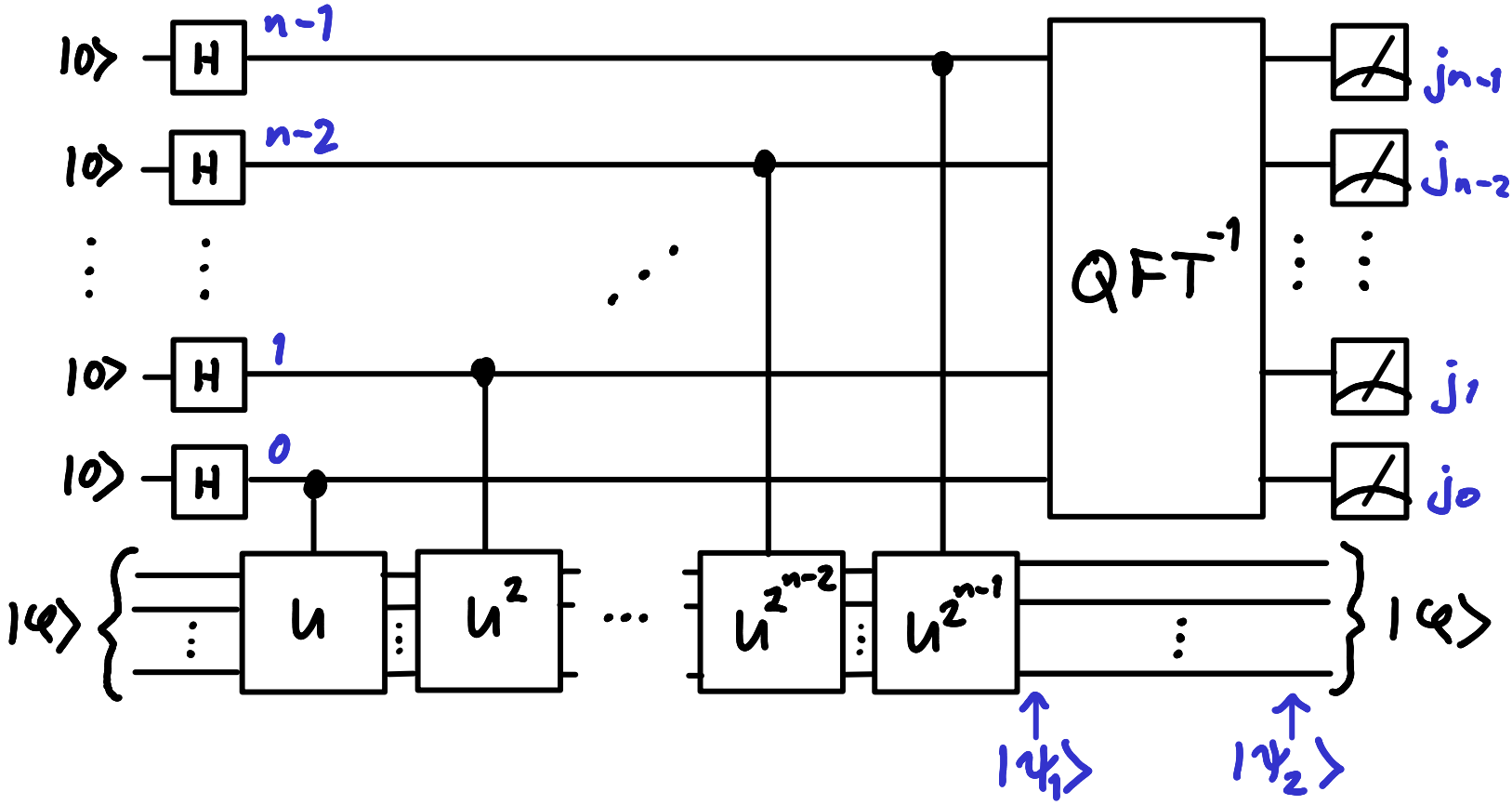
Problem (Phase Estimation)

Input: Black-box controlled unitary U on N qubits + 1 control qubit. State $|\psi\rangle$

Promise: $|\psi\rangle$ is an eigenstate of U: $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$

Output: Estimate θ to precision ε (I.e. output $\tilde{\theta}$ s.t. $|\tilde{\theta}-\theta| \leq \varepsilon$)

Algorithm (on $n+N$ qubits)



Output $\tilde{\theta} = 0.j_0 j_1 \cdots j_{n-1} = j/2^n$.

Note that $cU^k = (cU)^k$ so can build this given black-box c-U. However, $cU^{2^{n-1}}$ then requires $O(2^n)$ gates. So circuit is only efficient if $n = O(\log N)$, or if we have a U for which cU^k, $0 \le k \le 2^n$ can be implemented efficiently.

Analysis

Let $C = \prod_{j=0}^{n-1} c_j U^{2^j}$ = the cU part of the circuit

c_j controlled by j'th qubit

$$C|k\rangle|\psi\rangle$$

$$= \left(\prod_{j=0}^{n-1} c_j U^{2^j}\right) |k_0 k_1 \cdots k_{n-1}\rangle |\psi\rangle$$

$$= |k_0 k_1 \cdots k_{n-1}\rangle \prod_{j=0}^{n-1} U^{k_j 2^j} |\psi\rangle$$

$$= |k\rangle U^{\sum_j k_j 2^j} |\psi\rangle$$

$$= |k\rangle U^k |\psi\rangle$$

So C implements a controlled-U^k operation, with the value k set by the control register.

Now,

$$|\psi_1\rangle = C(H^{\otimes n} \otimes \mathbb{1})\,|0^n\rangle\,|\varphi\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_k C\,|k\rangle\,|\varphi\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_k |k\rangle\, U^k\,|\varphi\rangle$$

$$= \frac{1}{\sqrt{2^n}} \left(\sum_k e^{2\pi i k\theta}\,|k\rangle\right)|\varphi\rangle$$

recall $U|\varphi\rangle = e^{2\pi i\theta}|\varphi\rangle$

General version of phase kick-back phenomenon we saw previously.

$|\psi_1\rangle$ is a product state (i.e. not entangled) across $|k\rangle$ & $|\varphi\rangle$ registers & $|\varphi\rangle$ is returned unchanged.

Simple case: $\theta = \frac{j}{2^n}$, $j \in \{0, \ldots, 2^n - 1\}$

I.e. θ can be written exactly as a binary fraction $j = 0.j_0 j_1 \cdots j_{n-1}$ using at most n binary digits.

In this case,

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_k e^{2\pi i k j / 2^n} |k\rangle$$

(dropping the $|u\rangle$ part, since it's product & the circuit doesn't act further on it)

This should look familiar!

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_k e^{2\pi i k j / 2^n} |k\rangle$$

$$= QFT_{2^n} |j\rangle$$

Thus

$$|\psi_2\rangle = QFT_{2^n}^{-1} |\psi_1\rangle = |j\rangle$$

⟶ In this special case, phase estimation is <u>exact</u> ($\varepsilon = 0$).

Measuring output gives j (in binary) and $\theta = 0.j_0 j_1 \cdots j_{n-1} = j/2^n$.

General case:

Theorem (Phase Estimation)
If Phase Estimation outputs $\tilde{\theta} = 0.j_0 j_1 \cdots j_{n-1}$, then

(i) $\Pr(\tilde{\theta} = \text{closest } n\text{-digit binary approx. to } \theta) \geq 4/\pi^2$

(ii) $\Pr(|\tilde{\theta} - \theta| \geq \varepsilon) \leq O\left(\frac{1}{2^n \varepsilon}\right)$

From above, had

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \left(\sum_k e^{2\pi i k \theta} |k\rangle \right)$$ dropping $|\varphi\rangle$ again

So

$$|\psi_2\rangle = QFT_{2^n}^{-1} |\psi_1\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_k e^{2\pi i k \theta} \left(QFT_{2^n}^{-1} |k\rangle \right)$$

$$= \frac{1}{\sqrt{2^n}} \sum_k e^{2\pi i k \theta} \frac{1}{\sqrt{2^n}} \sum_j e^{-2\pi i j k / 2^n} |j\rangle$$

$$= \sum_j \left(\frac{1}{2^n} \sum_k e^{2\pi i k (\theta - j/2^n)} \right) |j\rangle$$

Thus

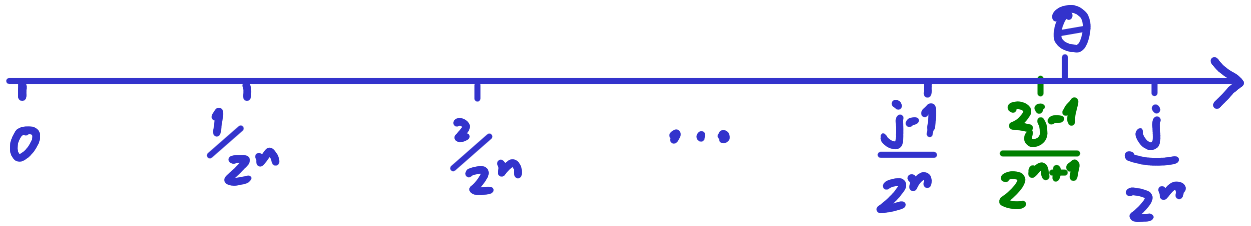
$$\Pr(\text{measure } j) = \left| \frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{2\pi i k(\theta - j/2^n)} \right|^2$$

$$= \begin{cases} 1 & \theta = j/2^n \\ \frac{1}{2^{2n}} \left| \frac{e^{2\pi i (2^n\theta - j)} - 1}{e^{2\pi i (\theta - j/2^n)} - 1} \right|^2 & \text{otherwise} \end{cases}$$

already saw this case

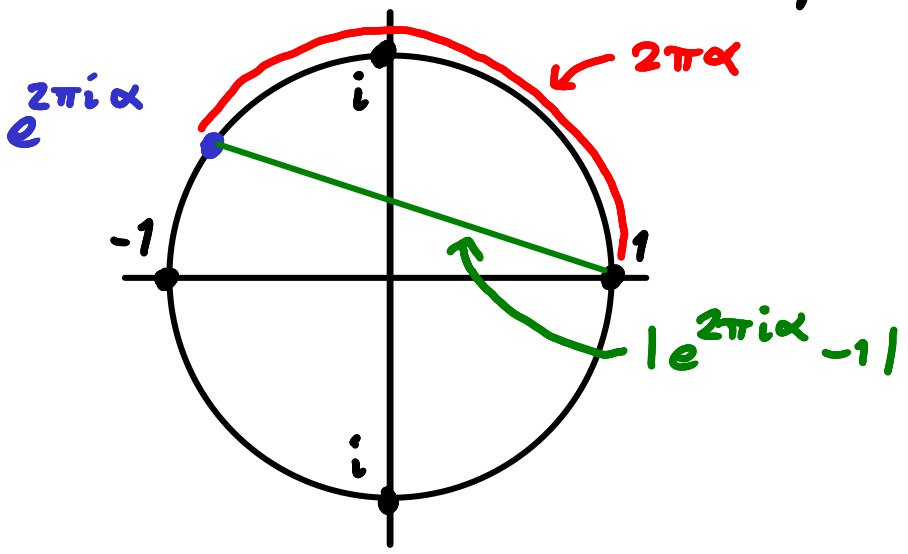
geometric series: $\sum_{k=0}^{n} r^k = \frac{r^n - 1}{r - 1}$

(i) Closest j to θ $\longrightarrow$ $\delta = |\theta - j/2^n| \leq \frac{1}{2^{n+1}}$:



$$\Pr(j) = \frac{1}{2^{2n}} \frac{|e^{2\pi i \delta 2^n} - 1|^2}{|e^{2\pi i \delta} - 1|^2}$$

Consider point $e^{2\pi i\alpha} - 1$ in complex plane:



$$1 \leq \frac{\text{arc length}}{\text{chord length}} = \frac{2\pi\alpha}{|e^{2\pi i\alpha} - 1|} \leq \frac{\pi}{2}$$

so

$$4\alpha \leq |e^{2\pi i\alpha} - 1| \leq 2\pi\alpha$$

$$\Rightarrow \Pr(j) = \frac{1}{2^{2n}} \frac{|e^{2\pi i\delta 2^n} - 1|^2}{|e^{2\pi i\delta} - 1|^2} \geq \frac{4}{\pi^2}.$$

using circle bound with $\alpha = \delta$ & $\alpha = \delta 2^n$

(ii) Let $\varepsilon = \theta - j/2^n$. We had

$$\Pr(j) = \frac{1}{2^{2n}} \left| \frac{e^{2\pi i (2^n\theta - j)} - 1}{e^{2\pi i(\theta - j/2^n)} - 1} \right|^2$$

$$= \frac{1}{2^{2n}} \frac{|e^{2\pi i \varepsilon 2^n} - 1|^2}{|e^{2\pi i \varepsilon} - 1|^2}$$

$$\leq \frac{1}{2^{2n}} \frac{4}{(4\varepsilon)^2}$$ using $|e^{2\pi i\alpha} - 1| \leq 2$ & circle bound with $\alpha = \varepsilon$

$$= \frac{1}{2^{2n+2}\varepsilon^2}$$

$$\Rightarrow \Pr(|\theta - j/2^n| \geq \varepsilon)$$

$$\leq \sum_{j : |\theta - j/2^n| \geq \varepsilon} \Pr(j) \leq \sum_{k=-\infty}^{\infty} \frac{1}{2^{2n+2}(\varepsilon + \frac{k}{2^n})^2}$$

$$\leq \frac{1}{4} \int_{-\infty}^{\infty} \frac{dk}{(2^n\varepsilon + k)^2} = \frac{1}{2} \int_0^{\infty} \frac{dk}{(2^n\varepsilon + k)^2}$$

$$= \frac{1}{2} \int_{2^n\varepsilon}^{\infty} \frac{dk'}{k'^2} = \frac{1}{2^{n+1}\varepsilon} \quad \square$$

$\longrightarrow$ To achieve phase estimation to precision ε with success probability $1-\delta$, need $n = O\left(\log \frac{1}{\varepsilon\delta}\right)$.

Like the QFT itself, the quantum phase estimation algorithm (QPE) is still a "quantum subroutine", not a fully fledged algorithm.

Any use of QPE to solve a bona fide computional problem must address two issues:

1. Circuit requires $O(2^n)$ gates (to implement cU^{2^n}) so can only afford $n = O(\log N) \longrightarrow \varepsilon, \delta = O\left(\frac{1}{\log N}\right)$ for efficient (i.e. poly(N)) circuit.

   Unless we have a particular U for which there is a more efficient way of implementing cU^{2^k}.

2. Algorithm requires eigenstate $|\psi\rangle$ as input, which may well be as hard to construct as finding its eigenvalue (the phase) in the first place.

   Sometimes, constructing superposition over eigenstates may be easier than constructing a single one $\longrightarrow$ gives following variant:

Problem (Phase Estimation variant)

Input: Black-box controlled unitary U on N qubits + 1 control qubit. State $|\psi\rangle$.

Promise: $|\psi\rangle = \sum_i \sqrt{p_i}\,|\psi_i\rangle$ where $U|\psi_i\rangle = e^{2\pi i \theta_i}|\psi_i\rangle$

Output: Estimate θ_i chosen at random from distribution p_i to precision ε.

Algorithm

Exactly the same as before!

Analysis

Exercise: Show this works.