EURO

Fundamental Quantum Computing Algorithms and Their Implementation in Qiskit

Michal Belina

IT4Innovations
VSB - Technical University of Ostrava

18. - 19.3. 2025

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Aims of the training

1. Familiarize you with the quantum computing
2. Show the possible advantage of quantum computers on specific tasks

Table of Contents

Timetable of day 2

9:00-10:30	Grover's algorithm
10:30-10:45	Break
10:45-12:00	Quantum Fourier Transform
12:00-13:00	Lunch Break
13:00-14:00	Quantum Phase Estimation
14:00-14:15	Break
14:15-15:45	Shor's algorithm
15:45	Q&A and Closing the event

Probably we will end sooner.

Table of Contents

Unstructured search

Let $\Sigma = \{0, 1\}$ denote the binary alphabet (throughout the lesson).
Suppose we're given a function

$$\boxed{f : \Sigma^n \rightarrow \Sigma}$$

that we can *compute efficiently.*
Our goal is to find a *solution*, which is a binary string $x \in \Sigma^n$ for which $f(x) = 1$.

Search

Input: $f : \Sigma^n \rightarrow \Sigma$
Output: a string $x \in \Sigma^n$ satisfying $f(x) = 1$, or "no solution" if no such strings exist.

This is *unstructured* search because f is arbitrary — there's *no promise* and we can't rely on it having a structure that makes finding solutions easy.

Search

Input: $f : \Sigma^n \to \Sigma$
Output: a string $x \in \Sigma^n$ satisfying $f(x) = 1$, or "no solution" if no such strings exist.

Hereafter let us write

$$\boxed{N = 2^n}$$

By iterating through all $x \in \Sigma^n$ and evaluating f on each one, we can solve Search with N queries.

This is the best we can do with a *deterministic* algorithm.

Probabilistic algorithms offer minor improvements, but still require a number of queries linear in N.

Grover's algorithm is a *quantum algorithm* for Search requiring $O(\sqrt{N})$ queries.

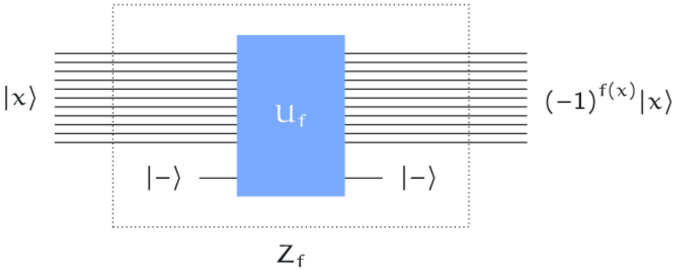
Phase query gates

We assume that we have access to the function $f : \Sigma^n \rightarrow \Sigma$ through a query gate:

$$\boxed{\mathbf{U_f} : |a\rangle|x\rangle \mapsto |a \oplus f(x)\rangle|x\rangle \quad (\text{for all } a \in \Sigma \text{ and } x \in \Sigma^n)}$$

(We can build a circuit for $\mathbf{U_f}$ given a Boolean circuit for f.)
The *phase query gate* for f operates like this:

$$\boxed{\mathbf{Z_f} : |x\rangle \mapsto (-1)^{f(x)}|x\rangle \quad (\text{for all } x \in \Sigma^n)}$$



Phase query gates

The *phase query gate* for f operates like this:

$$\boxed{\mathbf{Z_f} : |x\rangle \mapsto (-1)^{f(x)}|x\rangle \quad (\text{for all } x \in \Sigma^n)}$$

We're also going to need a phase query gate for the n-bit OR function:

$$\mathrm{OR}(x) = \begin{cases} 0 & x = 0^n \\ 1 & x \neq 0^n \end{cases} \quad (\text{for all } x \in \Sigma^n)$$

$$Z_{\mathrm{OR}}|x\rangle = \begin{cases} |x\rangle & x = 0^n \\ -|x\rangle & x \neq 0^n \end{cases} \quad (\text{for all } x \in \Sigma^n)$$

Grover's algorithm

1. *Initialize*: set n qubits to the state $H^{\otimes n}|0^n\rangle$.
2. *Iterate*: apply the *Grover operation* t times (for t to be specified later).
3. *Measure*: a standard basis measurement yields a candidate solution.

The Grover operation is defined like this:

$$G = H^{\otimes n} Z_{\mathrm{OR}} H^{\otimes n} Z_f$$

Z_f is the phase query gate for f and Z_{OR} is the phase query gate for the n-bit OR function.

Algorithm description

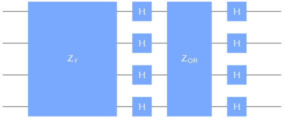
Grover's algorithm

1. *Initialize*: set n qubits to the state $H^{\otimes n}|0^n\rangle$.
2. *Iterate*: apply the *Grover operation* t times (for t to be specified later).
3. *Measure*: a standard basis measurement yields a candidate solution.

The Grover operation is defined like this:

$$G = H^{\otimes n} Z_{\mathrm{OR}} H^{\otimes n} Z_f$$

Z_f is the phase query gate for f and Z_{OR} is the phase query gate for the n-bit OR function.



Grover's algorithm

1. *Initialize*: set n qubits to the state $H^{\otimes n}|0^n\rangle$.
2. *Iterate*: apply the *Grover operation* t times (for t to be specified later).
3. *Measure*: a standard basis measurement yields a candidate solution.

The Grover operation is defined like this:

$$G = H^{\otimes n} Z_{\mathrm{OR}} H^{\otimes n} Z_f$$

Z_f is the phase query gate for f and Z_{OR} is the phase query gate for the n-bit OR function.

A typical way that Grover's algorithm can be applied:

1. Choose the number of iterations t (next section).
2. Run Grover's algorithm with t iterations to get a candidate solution x.
3. Check the solution. If $f(x) = 1$ then output x, otherwise either run Grover's algorithm again (possibly with a different t) or report "no solutions."

Solutions and non-solutions

We'll refer to the n qubits being used for Grover's algorithm as a register Q.
We're interested in what happens when Q is initialized to the state $H^{\otimes n}|0^n\rangle$ and the Grover operation G is performed iteratively.

$$G = H^{\otimes n} Z_{\mathrm{OR}} H^{\otimes n} Z_f$$

These are the sets of non-solutions and solutions:

$$\mathcal{A}_0 = \{x \in \Sigma^n : f(x) = 0\}$$
$$\mathcal{A}_1 = \{x \in \Sigma^n : f(x) = 1\}$$

We will be interested in *uniform superpositions* over these sets:

$$|\mathcal{A}_0\rangle = \frac{1}{\sqrt{|\mathcal{A}_0|}} \sum_{x \in \mathcal{A}_0} |x\rangle$$
$$|\mathcal{A}_1\rangle = \frac{1}{\sqrt{|\mathcal{A}_1|}} \sum_{x \in \mathcal{A}_1} |x\rangle$$

Analysis: basic idea

$$\mathcal{A}_0 = \{x \in \Sigma^n : f(x) = 0\} \quad \mathcal{A}_1 = \{x \in \Sigma^n : f(x) = 1\}$$
$$|\mathcal{A}_0\rangle = \frac{1}{\sqrt{|\mathcal{A}_0|}} \sum_{x \in \mathcal{A}_0} |x\rangle \quad |\mathcal{A}_1\rangle = \frac{1}{\sqrt{|\mathcal{A}_1|}} \sum_{x \in \mathcal{A}_1} |x\rangle$$

The register **Q** is first initialized to this state:

$$|u\rangle = H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \Sigma^n} |x\rangle$$

This state is contained in the subspace spanned by $|\mathcal{A}_0\rangle$ and $|\mathcal{A}_1\rangle$:

$$|u\rangle = \sqrt{\frac{|\mathcal{A}_0|}{N}}|\mathcal{A}_0\rangle + \sqrt{\frac{|\mathcal{A}_1|}{N}}|\mathcal{A}_1\rangle$$

The state of **Q** *remains in this subspace* after every application of the Grover operation G.

Action of the Grover operation

We can better understand the Grover operation by splitting it into two parts:

$$G = \left(H^{\otimes n} Z_{\mathrm{OR}} H^{\otimes n}\right)\left(Z_f\right)$$

1. Recall that Z_f is defined like this:

$$Z_f|x\rangle = (-1)^{f(x)}|x\rangle \quad (\text{for all } x \in \Sigma^n)$$

Its action on $|\mathcal{A}_0\rangle$ and $|\mathcal{A}_1\rangle$ is simple:

$$Z_f|\mathcal{A}_0\rangle = |\mathcal{A}_0\rangle$$
$$Z_f|\mathcal{A}_1\rangle = -|\mathcal{A}_1\rangle$$

Action of the Grover operation

We can better understand the Grover operation by splitting it into two parts:

$$G = \left(H^{\otimes n} Z_{\mathrm{OR}} H^{\otimes n}\right)(Z_f)$$

2. The operation Z_{OR} is defined like this:

$$Z_{\mathrm{OR}}|x\rangle = \begin{cases} |x\rangle & x = 0^n \\ -|x\rangle & x \neq 0^n \end{cases} \quad \text{(for all } x \in \Sigma^n\text{)}$$

Here's an alternative way to express Z_{OR}:

$$Z_{\mathrm{OR}} = 2|0^n\rangle\langle 0^n| - I$$

Using this expression, we can write $H^{\otimes n} Z_{\mathrm{OR}} H^{\otimes n}$ like this:

$$H^{\otimes n} Z_{\mathrm{OR}} H^{\otimes n} = H^{\otimes n}(2|0^n\rangle\langle 0^n| - I)H^{\otimes n} = 2|u\rangle\langle u| - I$$

Action of the Grover operation

$$Z_f|\mathcal{A}_0\rangle = |\mathcal{A}_0\rangle$$
$$Z_f|\mathcal{A}_1\rangle = -|\mathcal{A}_1\rangle$$
$$|u\rangle = \sqrt{\frac{|\mathcal{A}_0|}{N}}|\mathcal{A}_0\rangle + \sqrt{\frac{|\mathcal{A}_1|}{N}}|\mathcal{A}_1\rangle$$

$$\begin{aligned}
G|\mathcal{A}_0\rangle &= (2|u\rangle\langle u| - I)Z_f|\mathcal{A}_0\rangle \\
&= (2|u\rangle\langle u| - I)|\mathcal{A}_0\rangle \\
&= 2\sqrt{\frac{|\mathcal{A}_0|}{N}}|u\rangle - |\mathcal{A}_0\rangle \\
&= 2\sqrt{\frac{|\mathcal{A}_0|}{N}}\left(\sqrt{\frac{|\mathcal{A}_0|}{N}}|\mathcal{A}_0\rangle + \sqrt{\frac{|\mathcal{A}_1|}{N}}|\mathcal{A}_1\rangle\right) - |\mathcal{A}_0\rangle \\
&= \frac{|\mathcal{A}_0| - |\mathcal{A}_1|}{N}|\mathcal{A}_0\rangle + \frac{2\sqrt{|\mathcal{A}_0||\mathcal{A}_1|}}{N}|\mathcal{A}_1\rangle
\end{aligned}$$

Action of the Grover operation

$$Z_f|\mathcal{A}_0\rangle = |\mathcal{A}_0\rangle$$
$$Z_f|\mathcal{A}_1\rangle = -|\mathcal{A}_1\rangle$$
$$|u\rangle = \sqrt{\frac{|\mathcal{A}_0|}{N}}|\mathcal{A}_0\rangle + \sqrt{\frac{|\mathcal{A}_1|}{N}}|\mathcal{A}_1\rangle$$

$$G|\mathcal{A}_0\rangle = \frac{|\mathcal{A}_0| - |\mathcal{A}_1|}{N}|\mathcal{A}_0\rangle + \frac{2\sqrt{|\mathcal{A}_0||\mathcal{A}_1|}}{N}|\mathcal{A}_1\rangle$$

$$\begin{aligned}
G|\mathcal{A}_1\rangle &= (2|u\rangle\langle u| - I)Z_f|\mathcal{A}_1\rangle \\
&= (1 - 2|u\rangle\langle u|)|\mathcal{A}_1\rangle \\
&= |\mathcal{A}_1\rangle - 2\sqrt{\frac{|\mathcal{A}_1|}{N}}|u\rangle \\
&= |\mathcal{A}_1\rangle - 2\sqrt{\frac{|\mathcal{A}_0|}{N}}\left(\sqrt{\frac{|\mathcal{A}_0|}{N}}|\mathcal{A}_0\rangle + \sqrt{\frac{|\mathcal{A}_1|}{N}}|\mathcal{A}_1\rangle\right)
\end{aligned}$$

Action of the Grover operation

$$Z_f|\mathcal{A}_0\rangle = |\mathcal{A}_0\rangle$$
$$Z_f|\mathcal{A}_1\rangle = -|\mathcal{A}_1\rangle$$
$$|u\rangle = \sqrt{\frac{|\mathcal{A}_0|}{N}}|\mathcal{A}_0\rangle + \sqrt{\frac{|\mathcal{A}_1|}{N}}|\mathcal{A}_1\rangle$$

$$G|\mathcal{A}_0\rangle = \frac{|\mathcal{A}_0| - |\mathcal{A}_1|}{N}|\mathcal{A}_0\rangle + \frac{2\sqrt{|\mathcal{A}_0||\mathcal{A}_1|}}{N}|\mathcal{A}_1\rangle$$
$$G|\mathcal{A}_1\rangle = -\frac{2\sqrt{|\mathcal{A}_0||\mathcal{A}_1|}}{N}|\mathcal{A}_0\rangle + \frac{|\mathcal{A}_0| - |\mathcal{A}_1|}{N}|\mathcal{A}_1\rangle$$

The action of G on $\text{span}\{|\mathcal{A}_0\rangle, |\mathcal{A}_1\rangle\}$ can be described by a 2×2 matrix:

$$M = \begin{pmatrix} \frac{|\mathcal{A}_0|-|\mathcal{A}_1|}{N} & -\frac{2\sqrt{|\mathcal{A}_0||\mathcal{A}_1|}}{N} \\ \frac{2\sqrt{|\mathcal{A}_0||\mathcal{A}_1|}}{N} & \frac{|\mathcal{A}_0|-|\mathcal{A}_1|}{N} \end{pmatrix}$$

Rotation by an angle

The action of G on $\text{span}\{|\mathcal{A}_0\rangle, |\mathcal{A}_1\rangle\}$ can be described by a 2×2 matrix:

$$M = \begin{pmatrix} \frac{|\mathcal{A}_0|-|\mathcal{A}_1|}{N} & -\frac{2\sqrt{|\mathcal{A}_0||\mathcal{A}_1|}}{N} \\ \frac{2\sqrt{|\mathcal{A}_0||\mathcal{A}_1|}}{N} & \frac{|\mathcal{A}_0|-|\mathcal{A}_1|}{N} \end{pmatrix} = \left(\begin{pmatrix} \sqrt{\frac{|\mathcal{A}_0|}{N}} & -\sqrt{\frac{|\mathcal{A}_1|}{N}} \\ \sqrt{\frac{|\mathcal{A}_1|}{N}} & \sqrt{\frac{|\mathcal{A}_0|}{N}} \end{pmatrix} \right)^2$$

This is a *rotation* matrix.

$$\begin{pmatrix} \sqrt{\frac{|\mathcal{A}_0|}{N}} & -\sqrt{\frac{|\mathcal{A}_1|}{N}} \\ \sqrt{\frac{|\mathcal{A}_1|}{N}} & \sqrt{\frac{|\mathcal{A}_0|}{N}} \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \qquad \theta = \sin^{-1}\left(\sqrt{\frac{|\mathcal{A}_1|}{N}}\right)$$

$$M = \begin{pmatrix} \cos(2\theta) & -\sin(2\theta) \\ \sin(2\theta) & \cos(2\theta) \end{pmatrix}$$

Rotation by an angle

$$M = \begin{pmatrix} \cos(2\theta) & -\sin(2\theta) \\ \sin(2\theta) & \cos(2\theta) \end{pmatrix} \quad \theta = \sin^{-1}\left(\sqrt{\frac{|\mathcal{A}_1|}{N}}\right)$$

After the initialization step, this is the state of the register **Q**:

$$|u\rangle = \sqrt{\frac{|\mathcal{A}_0|}{N}}|\mathcal{A}_0\rangle + \sqrt{\frac{|\mathcal{A}_1|}{N}}|\mathcal{A}_1\rangle = \cos(\theta)|\mathcal{A}_0\rangle + \sin(\theta)|\mathcal{A}_1\rangle$$

Each time the Grover operation G is performed, the state of **Q** is rotated by an angle 2θ:

$$\begin{aligned} |u\rangle &= \cos(\theta)|\mathcal{A}_0\rangle + \sin(\theta)|\mathcal{A}_1\rangle \\ G|u\rangle &= \cos(3\theta)|\mathcal{A}_0\rangle + \sin(3\theta)|\mathcal{A}_1\rangle \\ G^2|u\rangle &= \cos(5\theta)|\mathcal{A}_0\rangle + \sin(5\theta)|\mathcal{A}_1\rangle \\ &\vdots \\ G^t|u\rangle &= \cos((2t+1)\theta)|\mathcal{A}_0\rangle + \sin((2t+1)\theta)|\mathcal{A}_1\rangle \end{aligned}$$

Geometric picture

Main idea

The operation $G = H^{\otimes n} Z_{OR} H^{\otimes n} Z_f$ is a composition of *two reflections:*

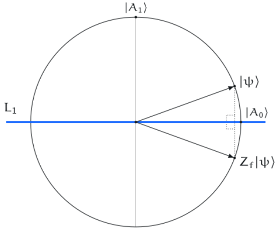
$$Z_f \quad \text{and} \quad H^{\otimes n} Z_{OR} H^{\otimes n}$$

Composing two reflections yields a *rotation.*

1. Recall that Z_f has this action on the 2-dimensional space spanned by $|A_0\rangle$ and $|A_1\rangle$:

$$Z_f|A_0\rangle = |A_0\rangle$$
$$Z_f|A_1\rangle = -|A_1\rangle$$

This is a *reflection* about the line L_1 parallel to $|A_0\rangle$.



The operation $G = H^{\otimes n} Z_{OR} H^{\otimes n} Z_f$ is a composition of *two reflections:*

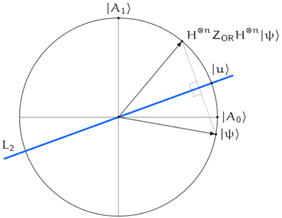
$$Z_f \quad \text{and} \quad H^{\otimes n} Z_{OR} H^{\otimes n}$$

Composing two reflections yields a *rotation.*

2. The operation $H^{\otimes n} Z_{OR} H^{\otimes n}$ can be expressed like this:

$$H^{\otimes n} Z_{OR} H^{\otimes n} = 2|u\rangle\langle u| - \mathbb{I}$$

Again this is a *reflection*, this time about the line L_2 parallel to $|u\rangle$.



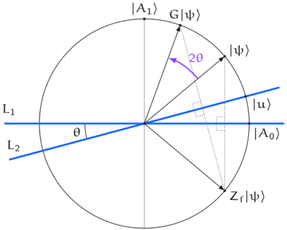
Main idea

The operation $G = H^{\otimes n} Z_{OR} H^{\otimes n} Z_f$ is a composition of *two reflections:*

$$Z_f \quad \text{and} \quad H^{\otimes n} Z_{OR} H^{\otimes n}$$

Composing two reflections yields a *rotation.*

When we compose two reflections, we obtain a *rotation* by twice the angle between the lines of reflection.



Setting the target

Consider any quantum state of this form:

$$\alpha|A_0\rangle + \beta|A_1\rangle$$

Measuring yields a solution $x \in A_1$ with probability $|\beta|^2$.

$$\alpha|A_0\rangle + \beta|A_1\rangle = \frac{\alpha}{\sqrt{|A_0|}} \sum_{x \in A_0} |x\rangle + \frac{\beta}{\sqrt{|A_1|}} \sum_{x \in A_1} |x\rangle$$

$$p(x) = \begin{cases} \frac{|\alpha|^2}{|A_0|}, & x \in A_0 \\ \frac{|\beta|^2}{|A_1|}, & x \in A_1 \end{cases}$$

$$\Pr(\text{outcome is in } A_1) = \sum_{x \in A_1} p(x) = |\beta|^2$$

Setting the target

Consider any quantum state of this form:

$$\alpha|A_0\rangle + \beta|A_1\rangle$$

Measuring yields a solution $x \in A_1$ with probability $|\beta|^2$.

The state of Q after t iterations in Grover's algorithm:

$$\cos((2t+1)\theta)|A_0\rangle + \sin((2t+1)\theta)|A_1\rangle \qquad \theta = \sin^{-1}\left(\sqrt{\frac{|A_1|}{N}}\right)$$

Measuring after t iterations gives an outcome $x \in A_1$ with probability:

$$\sin^2((2t+1)\theta)$$

We wish to maximize this probability—so we may view that $|A_1\rangle$ is our *target state.*

Setting the target

The state of Q after t iterations in Grover's algorithm:

$$\cos((2t+1)\theta)|A_0\rangle + \sin((2t+1)\theta)|A_1\rangle \qquad \theta = \sin^{-1}\left(\sqrt{\tfrac{|A_1|}{N}}\right)$$

Measuring after t iterations gives an outcome $x \in A_1$ with probability:

$$\sin^2((2t+1)\theta)$$

To make this probability close to 1 and minimize t, we will aim for:

$$(2t+1)\theta \approx \tfrac{\pi}{2} \quad \Leftrightarrow \quad t \approx \tfrac{\pi}{4\theta} - \tfrac{1}{2} \quad \text{closest integer} \quad \Rightarrow \quad t = \left\lfloor \tfrac{\pi}{4\theta} \right\rfloor$$

Important considerations:

- t must be an integer
- θ depends on the number of solutions $s = |A_1|$

Unique search

$$(2t+1)\theta \approx \frac{\pi}{2} \quad \Leftrightarrow \quad t = \left\lfloor \frac{\pi}{4\theta} \right\rfloor$$

Unique search

Input: $f : \Sigma^n \to \Sigma$
Promise: There is exactly one string $z \in \Sigma^n$ for which $f(z) = 1$, with $f(x) = 0$ for all strings $x \neq z$
Output: The string z

For Unique search we have $s = |A_1| = 1$ and therefore:

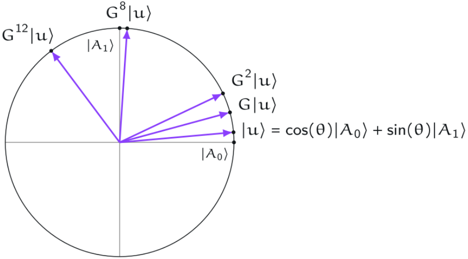
$$\theta = \sin^{-1}\left(\sqrt{\frac{1}{N}}\right) \approx \sqrt{\frac{1}{N}}$$

Substituting $\theta \approx 1/\sqrt{N}$ into our expression for t gives:

$$t \approx \left\lfloor \frac{\pi}{4\sqrt{N}} \right\rfloor \quad \Leftarrow \quad \mathcal{O}(\sqrt{N}) \text{ queries}$$

Example: N = 128

$$\theta = \sin^{-1}\left(\frac{1}{\sqrt{N}}\right) = 0.0885...$$
$$t = \left\lfloor \frac{\pi}{4\theta} \right\rfloor = 8$$



Unique search

$$\theta = \sin^{-1}\left(\sqrt{\frac{1}{N}}\right) \qquad t = \left\lfloor \frac{\pi}{4\theta} \right\rfloor$$

Measuring after t iterations gives the (unique) outcome $x \in A_1$ with probability:

$$p(N, 1) = \sin^2((2t + 1)\theta)$$

Success probabilities for Unique search

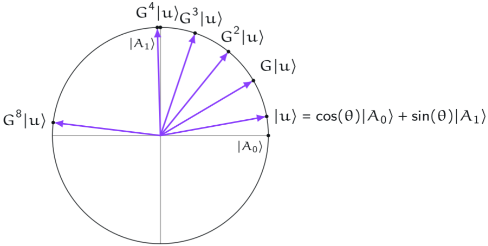
N	$p(N,1)$	N	$p(N,1)$
2	0.5	128	0.9956199
4	1.0	256	0.9999470
8	0.9453125	512	0.9994480
16	0.9613190	1024	0.9994612
32	0.9991823	2048	0.9999968
64	0.9965857	4096	0.9999453

It can be proved analytically that $p(N, 1) \geq 1 - \frac{1}{N}$.

Multiple solutions

Example: N = 128, s = 4

$$\theta = \sin^{-1}\left(\sqrt{\frac{s}{N}}\right) = 0.1777\cdots$$
$$t = \left\lfloor \frac{\pi}{4\theta} \right\rfloor = 4$$



Multiple solutions

$$\theta = \sin^{-1}\left(\sqrt{\tfrac{s}{N}}\right) \qquad t = \left\lfloor \tfrac{\pi}{4\theta} \right\rfloor$$

For every $s \in \{1, \ldots, N\}$, the probability $p(N, s)$ to find a solution satisfies:

$$p(N, s) \geq \max\left\{1 - \tfrac{s}{N}, \tfrac{s}{N}\right\}$$

Number of queries

$$\theta = \sin^{-1}\left(\sqrt{\tfrac{s}{N}}\right) \qquad t = \left\lfloor \tfrac{\pi}{4\theta} \right\rfloor$$

Each iteration of Grover's algorithm requires 1 query (or evaluation of f). How does the number of queries t depend on N and s?

$$\begin{gathered}
\sin^{-1}(x) \geq x \quad (\text{for every } x \in [0,1]) \\
\theta = \sin^{-1}\left(\sqrt{\tfrac{s}{N}}\right) \geq \sqrt{\tfrac{s}{N}} \\
t \leq \tfrac{\pi}{4\theta} \leq \tfrac{\pi}{4}\sqrt{\tfrac{N}{s}} \\
t = \mathcal{O}\left(\sqrt{\tfrac{N}{s}}\right)
\end{gathered}$$

Unknown number of solutions

What do we do if we don't know the number of solutions in advance?

A simple approach

Choose the number of iterations $t \in \{1, \ldots, \lfloor \pi\sqrt{N}/4 \rfloor\}$ *uniformly at random.*

- The probability to find a solution (if one exists) will be at least 40%. (Repeat several times to boost success probability.)
- The number of queries (or evaluations of f) is $O(\sqrt{N})$.

A more sophisticated approach

1. Set $T = 1$.
2. Run Grover's algorithm with $t \in \{1, \ldots, T\}$ chosen uniformly at random.
3. If a solution is found, output it and stop. Otherwise, increase T and return to step 2 (or report "no solut

- The rate of increase of T must be carefully balanced: slower rates require more queries, higher rates decrease success probability. $T \leftarrow \left\lceil \frac{5}{4} T \right\rceil$ works.
- If the number of solutions is $s \geq 1$, then the number of queries (or evaluations of f) required is $O(\sqrt{N/s})$. If there are no solutions, $O(\sqrt{N})$ queries are required.

Table of Contents

Spectral theorem for unitary matrices

The *spectral theorem* is an important fact in linear algebra. Here is a statement of a special case of this theorem, for *unitary matrices.*

Spectral theorem for unitary matrices

Suppose U is an $N \times N$ unitary matrix.
There exists an orthonormal basis $\{|\psi_1\rangle, \ldots, |\psi_N\rangle\}$ of vectors along with complex numbers

$$\lambda_1 = e^{2\pi i \theta_1}, \ldots, \lambda_N = e^{2\pi i \theta_N}$$

such that

$$U = \sum_{k=1}^{N} \lambda_k |\psi_k\rangle\langle\psi_k|$$

Spectral theorem for unitary matrices

The *spectral theorem* is an important fact in linear algebra. Here is a statement of a special case of this theorem, for *unitary matrices.*

Spectral theorem for unitary matrices

Suppose U is an $N \times N$ unitary matrix.
There exists an orthonormal basis $\{|\psi_1\rangle, \dots, |\psi_N\rangle\}$ of vectors along with complex numbers

$$\lambda_1 = e^{2\pi i\theta_1}, \dots, \lambda_N = e^{2\pi i\theta_N}$$

such that

$$U = \sum_{k=1}^{N} \lambda_k |\psi_k\rangle\langle\psi_k|$$

Each vector $|\psi_k\rangle$ is an *eigenvector* of U having *eigenvalue* λ_k:

$$U|\psi_k\rangle = \lambda_k|\psi_k\rangle = e^{2\pi i\theta_k}|\psi_k\rangle$$

Phase estimation problem

In the phase estimation problem, we're given two things:

1. A description of a *unitary quantum circuit* on n qubits.
2. An n-qubit *quantum state* $|\psi\rangle$.

We're *promised* that $|\psi\rangle$ is an eigenvector of the unitary operation U described by the circuit, and our goal is to approximate the corresponding eigenvalue.

Phase estimation problem

Input: A unitary quantum circuit for an n-qubit operation U and an n-qubit quantum state $|\psi\rangle$
Promise: $|\psi\rangle$ is an eigenvector of U
Output: An approximation to the number $\theta \in [0, 1]$ satisfying

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

In the phase estimation problem, we're given two things:

1. A description of a *unitary quantum circuit* on n qubits.
2. An n-qubit *quantum state* $|\psi\rangle$.

We're *promised* that $|\psi\rangle$ is an eigenvector of the unitary operation U described by the circuit, and our goal is to approximate the corresponding eigenvalue.

Phase estimation problem

Input: A unitary quantum circuit for an n-qubit operation U and an n-qubit quantum state $|\psi\rangle$
Promise: $|\psi\rangle$ is an eigenvector of U
Output: An approximation to the number $\theta \in [0, 1]$ satisfying

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

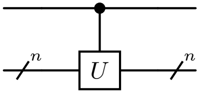
We can approximate θ by a fraction:

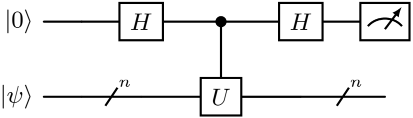
$$\theta \approx \frac{y}{2^m}$$

for $y \in \{0, 1, \ldots, 2^m - 1\}$. This approximation is taken *modulo 1*.

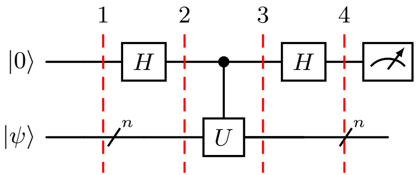
Warm-up: using the phase kickback

Given a circuit for U, we can create a circuit for a controlled-U operation:





Warm-up: using the phase kickback II

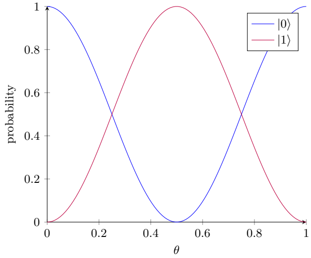


$$
\begin{aligned}
|\pi_1\rangle &= |\psi\rangle|0\rangle \\
|\pi_2\rangle &= \tfrac{1}{\sqrt{2}}|\psi\rangle|0\rangle + \tfrac{1}{\sqrt{2}}|\psi\rangle|1\rangle \\
|\pi_3\rangle &= \tfrac{1}{\sqrt{2}}\,|\psi\rangle\,|0\rangle + \tfrac{1}{\sqrt{2}}(U\,|\psi\rangle)\,|1\rangle = |\psi\rangle \otimes \left(\tfrac{1}{\sqrt{2}}\,|0\rangle + \tfrac{e^{2\pi i\theta}}{\sqrt{2}}\,|1\rangle\right) \\
|\pi_4\rangle &= |\psi\rangle \otimes \left(\tfrac{1+e^{2\pi i\theta}}{2}|0\rangle + \tfrac{1-e^{2\pi i\theta}}{2}|1\rangle\right)
\end{aligned}
$$

$$|\pi_4\rangle = |\psi\rangle \otimes \left(\frac{1+e^{2\pi i\theta}}{2}|0\rangle + \frac{1-e^{2\pi i\theta}}{2}|1\rangle \right)$$

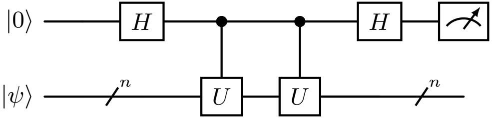
$$p_0 = \left| \frac{1+e^{2\pi i\theta}}{2} \right|^2 = \cos^2(\pi\theta) \quad p_1 = \left| \frac{1-e^{2\pi i\theta}}{2} \right|^2 = \sin^2(\pi\theta)$$

Measuring the top qubit yields the outcomes 0 and 1 with these probabilities:

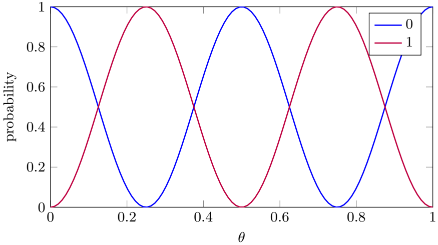


Iterating the unitary operation

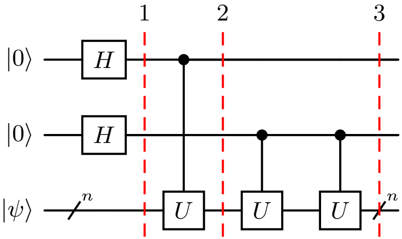
How can we learn more about θ? One possibility is to apply the controlled-U operation twice (or multiple times):



Performing the controlled-U operation twice has the effect of squaring the eigenvalue:



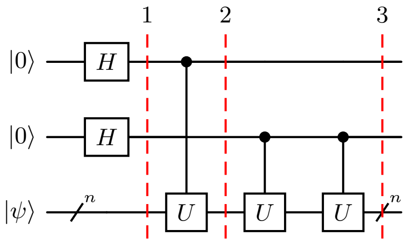
Two control qubits



$$|\pi_1\rangle = |\psi\rangle \otimes \frac{1}{2} \sum_{a_0=0}^{1} \sum_{a_1=0}^{1} |a_1 a_0\rangle$$

$$|\pi_2\rangle = |\psi\rangle \otimes \frac{1}{2} \sum_{a_0=0}^{1} \sum_{a_1=0}^{1} e^{2\pi i a_0 \theta} |a_1 a_0\rangle$$

Two control qubits



$$|\pi_3\rangle = |\psi\rangle \otimes \frac{1}{2} \sum_{a_0=0}^{1} \sum_{a_1=0}^{1} e^{2\pi i (2a_1 + a_0)\theta} |a_1 a_0\rangle \tag{1}$$

$$= |\psi\rangle \otimes \frac{1}{2} \sum_{x=0}^{3} e^{2\pi i x\theta} |x\rangle \tag{2}$$

Two control qubits

What can we learn about θ from this state? Suppose we're promised that $\theta = \frac{y}{4}$ for $y \in \{0,1,2,3\}$. Can we figure out which one it is?
Define a two-qubit state for each possibility:

$$|\Phi_y\rangle = \frac{1}{2}\sum_{x=0}^{3} e^{2\pi i \frac{xy}{4}} |x\rangle$$

$$|\Phi_0\rangle = \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|2\rangle + \frac{1}{2}|3\rangle \qquad |\Phi_1\rangle = \frac{1}{2}|0\rangle + \frac{i}{2}|1\rangle - \frac{1}{2}|2\rangle - \frac{i}{2}|3\rangle$$

$$|\Phi_2\rangle = \frac{1}{2}|0\rangle - \frac{1}{2}|1\rangle + \frac{1}{2}|2\rangle - \frac{1}{2}|3\rangle \qquad |\Phi_3\rangle = \frac{1}{2}|0\rangle - \frac{i}{2}|1\rangle - \frac{1}{2}|2\rangle + \frac{i}{2}|3\rangle$$

These vectors are *orthonormal*—so they can be discriminated perfectly by a projective measurement.

Two control qubits

Unitary Matrix Representation

$$V = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \tag{3}$$

Action of the Unitary Matrix

$$V\,|y\rangle = |\Phi_y\rangle \quad \text{(for every } y \in \{0,1,2,3\}\text{)} \tag{4}$$

Inverse Operation

We can identify y by performing the inverse of V and then a standard basis measurement:

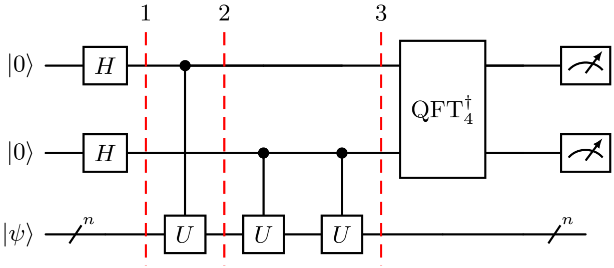
$$V^\dagger\,|\Phi_y\rangle = |y\rangle \quad \text{(for every } y \in \{0,1,2,3\}\text{)} \tag{5}$$

Table of Contents

Two-qubit phase estimation

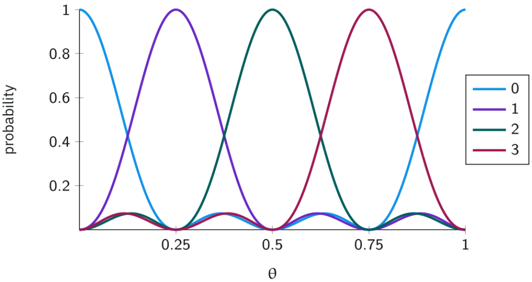
This matrix is associated with the *discrete Fourier transform* (for 4 dimensions). When we think about this matrix as a unitary operation, we call it the *quantum Fourier transform*.

The complete circuit for learning $y \in \{0, 1, 2, 3\}$ when $\theta = y/4$:



Two-qubit phase estimation

The outcome probabilities when we run the circuit, as a function of θ:



Quantum Fourier Transform

The quantum Fourier transform is defined for each positive integer N:

$$\mathrm{QFT}_N = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^{2\pi i \frac{xy}{N}} |x\rangle\langle y|$$

$$\mathrm{QFT}_N |y\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} e^{2\pi i \frac{xy}{N}} |x\rangle$$

$$\mathrm{QFT}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H$$

$$\mathrm{QFT}_3 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \frac{-1+i\sqrt{3}}{2} & \frac{-1-i\sqrt{3}}{2} \\ 1 & \frac{-1-i\sqrt{3}}{2} & \frac{-1+i\sqrt{3}}{2} \end{pmatrix}$$

Quantum Fourier Transform

The quantum Fourier transform is defined for each positive integer N as follows.

$$\mathrm{QFT}_N = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^{2\pi i \frac{xy}{N}} |x\rangle\langle y|$$

$$\mathrm{QFT}_N |y\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} e^{2\pi i \frac{xy}{N}} |x\rangle$$

$$\mathrm{QFT}_4 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

Quantum Fourier Transform

$$\mathrm{QFT}_8 = \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{1+i}{\sqrt{2}} & i & \frac{-1+i}{\sqrt{2}} & -1 & \frac{-1-i}{\sqrt{2}} & -i & \frac{1-i}{\sqrt{2}} \\ 1 & i & -1 & -i & 1 & i & -1 & -i \\ 1 & \frac{-1+i}{\sqrt{2}} & -i & \frac{1+i}{\sqrt{2}} & -1 & \frac{1-i}{\sqrt{2}} & i & \frac{-1-i}{\sqrt{2}} \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & \frac{-1-i}{\sqrt{2}} & -i & \frac{1-i}{\sqrt{2}} & -1 & \frac{1+i}{\sqrt{2}} & i & \frac{-1+i}{\sqrt{2}} \\ 1 & -i & -1 & i & 1 & -i & -1 & i \\ 1 & \frac{1-i}{\sqrt{2}} & -i & \frac{-1-i}{\sqrt{2}} & -1 & \frac{-1+i}{\sqrt{2}} & i & \frac{1+i}{\sqrt{2}} \end{pmatrix}$$

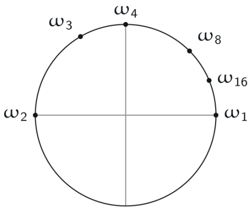
Quantum Fourier Transform

The quantum Fourier transform is defined for each positive integer N as follows.

$$\mathrm{QFT}_N = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^{2\pi i \frac{xy}{N}} |x\rangle\langle y| = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \omega_N^{xy} |x\rangle\langle y|$$

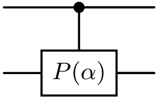
Useful shorthand notation:

$$\omega_N = e^{\frac{2\pi i}{N}} = \cos\left(\frac{2\pi}{N}\right) + i \sin\left(\frac{2\pi}{N}\right)$$

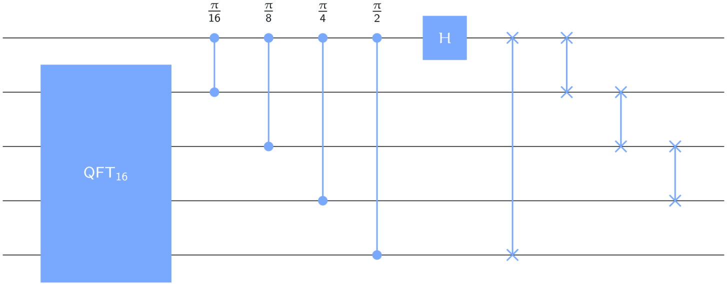


Circuits for the QFT

We can implement $\mathbf{QFT}_N$ efficiently with a quantum circuit when N is a power of 2.
The implementation makes use of **controlled-phase** gates:



$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\alpha} \end{pmatrix}$$



Cost analysis

Let s_m denote the number of gates we need for m qubits.

- For $m = 1$, a single Hadamard gate is required.
- For $m \geq 2$, these are the gates required:
  - s_{m-1} gates for the QFT on $m-1$ qubits
  - $m-1$ controlled phase gates
  - $m-1$ swap gates
  - 1 Hadamard gate

$$s_m = \begin{cases} 1 & m = 1 \\ s_{m-1} + 2m - 1 & m \geq 2 \end{cases}$$

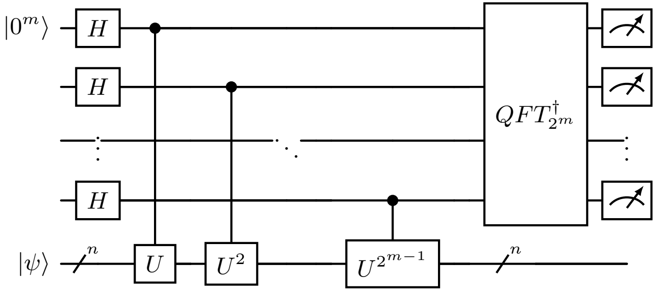
This is a ***recurrence relation*** with a closed-form solution:

$$s_m = \sum_{k=1}^{m} (2k-1) = m^2$$

Additional remarks:

- The number of swap gates can be reduced.
- Approximations to QFT_{2^m} can be done at lower cost (and lower depth).

The general phase-estimation procedure, for any choice of m:

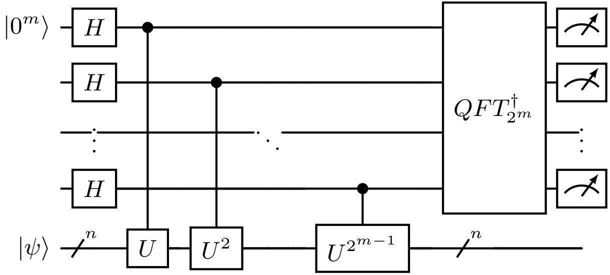


Warning

If we perform each U^k-operation by repeating a controlled-U operation k times, increasing the number of control qubits m comes at a ***high cost***.

Phase estimation procedure

The general phase-estimation procedure, for any choice of m:



$$|\pi\rangle = |\psi\rangle \otimes \frac{1}{2^m} \sum_{y=0}^{2^m-1} \sum_{x=0}^{2^m-1} e^{2\pi i x(\theta - y/2^m)} |y\rangle \tag{6}$$

$$p_y = \left| \frac{1}{2^m} \sum_{x=0}^{2^m-1} e^{2\pi i x(\theta - y/2^m)} \right|^2 \tag{7}$$

Phase estimation procedure

$$p_y = \left| \frac{1}{2^m} \sum_{x=0}^{2^m-1} e^{2\pi i x(\theta - y/2^m)} \right|^2 \tag{8}$$

Best approximations

Suppose $\frac{y}{2^m}$ is the ***best approximation*** to θ:

$$\left|\theta - \frac{y}{2^m}\right|_1 \leq 2^{-(m+1)}$$

Then the probability to measure y will be relatively high:

$$p_y \geq \frac{4}{\pi^2} \approx 0.405$$

Worse approximations

Suppose there's a ***better approximation*** to θ between $\frac{y}{2^m}$ and θ:

$$\left|\theta - \frac{y}{2^m}\right|_1 \geq 2^{-m}$$

Then the probability to measure y will be relatively low:

$$p_y \leq \frac{1}{4}$$

Phase Estimation Accuracy

To obtain an approximation $\frac{y}{2^m}$ that is ***very likely*** to satisfy

$$\left|\theta - \frac{y}{2^m}\right|_1 < 2^{-m}$$

we can run the phase estimation procedure using m control qubits ***several times*** and take y to be the ***mode*** of the outcomes.

(The eigenvector $|\psi\rangle$ is unchanged by the procedure and can be reused as many times as needed.)

Table of Contents

The order-finding problem

For each positive integer N, we define

$$\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$$

For instance, $\mathbb{Z}_1 = \{0\}$, $\mathbb{Z}_2 = \{0, 1\}$, $\mathbb{Z}_3 = \{0, 1, 2\}$, and so on.
We can view arithmetic operations on $\mathbb{Z}_N$ as being defined modulo N.

Example

Let $N = 7$. We have $3 \cdot 5 = 15$, which leaves a remainder of 1 when divided by 7.
This is often expressed like this:

$$3 \cdot 5 \equiv 1 \pmod{7}$$

We can also simply write $3 \cdot 5 = 1$ when it's clear we're working in $\mathbb{Z}_7$.
The elements $a \in \mathbb{Z}_N$ that satisfy $\gcd(a, N) = 1$ are special.

$$\mathbb{Z}_N^* = \{a \in \mathbb{Z}_N : \gcd(a, N) = 1\}$$
$$\mathbb{Z}_{21}^* = \{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$$

The order-finding problem

Fact

For every $a \in \mathbb{Z}_N^*$ there must exist a positive integer k such that $a^k = 1$. The smallest such k is called the ***order*** of a in $\mathbb{Z}_N^*$.

Example

For $N = 21$, these are the smallest powers for which this works:

$$\begin{array}{cccc} 1^1 = 1 & 5^6 = 1 & 11^6 = 1 & 17^6 = 1 \\ 2^6 = 1 & 8^2 = 1 & 13^2 = 1 & 19^6 = 1 \\ 4^3 = 1 & 10^6 = 1 & 16^3 = 1 & 20^2 = 1 \end{array}$$

Order-finding problem

Input: Positive integers a and N with $\gcd(a, N) = 1$.
Output: The smallest positive integer r such that $a^r \equiv 1 \pmod{N}$.

No efficient classical algorithm for this problem is known — an efficient algorithm for order-finding implies an efficient algorithm for integer factorization.

Order-finding by phase-estimation

To connect the order-finding problem to phase estimation, consider a system whose classical state set is $\mathbb{Z}_N$.
For a given element $a \in \mathbb{Z}_N^*$, define an operation as follows:

$$\mathcal{M}_a|x\rangle = |ax\rangle \quad (\text{for each } x \in \mathbb{Z}_N)$$

This is a ***unitary operation***—but only because $\gcd(a, N) = 1$!

Example

Let $N = 15$ and $a = 2$. The operation $\mathcal{M}_a$ has this action:

$$\begin{array}{lll}
\mathcal{M}_2|0\rangle = |0\rangle & \mathcal{M}_2|5\rangle = |10\rangle & \mathcal{M}_2|10\rangle = |5\rangle \\
\mathcal{M}_2|1\rangle = |2\rangle & \mathcal{M}_2|6\rangle = |12\rangle & \mathcal{M}_2|11\rangle = |7\rangle \\
\mathcal{M}_2|2\rangle = |4\rangle & \mathcal{M}_2|7\rangle = |14\rangle & \mathcal{M}_2|12\rangle = |9\rangle \\
\mathcal{M}_2|3\rangle = |6\rangle & \mathcal{M}_2|8\rangle = |1\rangle & \mathcal{M}_2|13\rangle = |11\rangle \\
\mathcal{M}_2|4\rangle = |8\rangle & \mathcal{M}_2|9\rangle = |3\rangle & \mathcal{M}_2|14\rangle = |13\rangle
\end{array}$$

Order-finding by phase-estimation

To connect the order-finding problem to phase estimation, consider a system whose classical state set is $\mathbb{Z}_N$.
For a given element $a \in \mathbb{Z}_N^*$, define an operation as follows:

$$\mathcal{M}_a|x\rangle = |ax\rangle \quad (\text{for each } x \in \mathbb{Z}_N)$$

This is a ***unitary operation***—but only because $\gcd(a, N) = 1$!

Main idea

The ***eigenvalues*** of $\mathcal{M}_a$ are closely connected with the ***order*** of a.

By approximating certain eigenvalues with enough precision using phase estimation, we'll be able to compute the order.

Eigenvectors and eigenvalues

This is an eigenvector of $\mathcal{M}_a$:

$$|\psi_0\rangle = \frac{|1\rangle + |a\rangle + \cdots + |a^{r-1}\rangle}{\sqrt{r}}$$

The associated eigenvalue is 1:

$$\mathcal{M}_a|\psi_0\rangle = \frac{|a\rangle + |a^2\rangle + \cdots + |a^r\rangle}{\sqrt{r}} = \frac{|a\rangle + \cdots + |a^{r-1}\rangle + |1\rangle}{\sqrt{r}} = |\psi_0\rangle$$

To identify more eigenvectors, first recall that

$$\omega_r = e^{2\pi i/r}$$

This is another eigenvector of $\mathcal{M}_a$:

$$|\psi_1\rangle = \frac{|1\rangle + \omega_r^{-1}|a\rangle + \cdots + \omega_r^{-(r-1)}|a^{r-1}\rangle}{\sqrt{r}}$$

Eigenvectors and eigenvalues

$$
\begin{aligned}
\mathcal{M}_a|\psi_1\rangle &= \frac{|a\rangle + \omega_r^{-1}|a^2\rangle + \cdots + \omega_r^{-(r-1)}|a^r\rangle}{\sqrt{r}} \\
&= \frac{\omega_r|1\rangle + |a\rangle + \omega_r^{-1}|a^2\rangle + \cdots + \omega_r^{-(r-2)}|a^{r-1}\rangle}{\sqrt{r}} \\
&= \omega_r \left(\frac{|1\rangle + \omega_r^{-1}|a\rangle + \omega_r^{-2}|a^2\rangle + \cdots + \omega_r^{-(r-1)}|a^{r-1}\rangle}{\sqrt{r}} \right) \\
&= \omega_r|\psi_1\rangle
\end{aligned}
$$

Additional eigenvectors can be identified by similar reasoning...
For each $j \in \{0, \ldots, r-1\}$, this is an eigenvector of $\mathcal{M}_a$:

$$
|\psi_j\rangle = \frac{|1\rangle + \omega_r^{-j}|a\rangle + \cdots + \omega_r^{-j(r-1)}|a^{r-1}\rangle}{\sqrt{r}}
$$

$$
\mathcal{M}_a|\psi_j\rangle = \omega_r^j|\psi_j\rangle
$$

A convenient eigenvector

$$|\psi_1\rangle = \frac{|1\rangle + \omega_r^{-1}|a\rangle + \cdots + \omega_r^{-(r-1)}|a^{r-1}\rangle}{\sqrt{r}}$$

$$\mathcal{M}_a|\psi_1\rangle = \omega_r|\psi_1\rangle = e^{2\pi i \frac{1}{r}}|\psi_1\rangle$$

Suppose we're given $|\psi_1\rangle$ as a quantum state. We can attempt to learn r as follows:

1. Perform phase estimation on the state $|\psi_1\rangle$ and a quantum circuit implementing $\mathcal{M}_a$. The outcome is an approximation $y/2^m \approx 1/r$.
2. Output $2^m/y$ rounded to the nearest integer:

$$\text{round}\left(\frac{2^m}{y}\right) = \left\lfloor \frac{2^m}{y} + \frac{1}{2} \right\rfloor$$

How much precision do we need to correctly determine r?

$$\left|\frac{y}{2^m} - \frac{1}{r}\right| \leq \frac{1}{2N^2} \quad \Rightarrow \quad \text{round}\left(\frac{2^m}{y}\right) = r$$

Choosing $m = 2\log(N) + 1$ in phase estimation makes such an approximation likely.

A random eigenvector

$$|\psi_j\rangle = \frac{|1\rangle + \omega_r^{-j}|a\rangle + \cdots + \omega_r^{-j(r-1)}|a^{r-1}\rangle}{\sqrt{r}}$$

$$\mathcal{M}_a|\psi_j\rangle = \omega_r^j|\psi_j\rangle = e^{2\pi i \frac{j}{r}}|\psi_j\rangle$$

Suppose we're given $|\psi_j\rangle$ as a quantum state for a ***random choice*** of $j \in \{0, \dots, r-1\}$. We can attempt to learn j/r as follows:

1. Perform phase estimation on the state $|\psi_j\rangle$ and a quantum circuit implementing $\mathcal{M}_a$. The outcome is an approximation $y/2^m \approx j/r$.
2. Among the fractions u/v in lowest terms satisfying $u, v \in \{0, \dots, N-1\}$ and $v \neq 0$, output the one closest to $y/2^m$. This can be done efficiently using the ***continued fraction algorithm.***

How much precision do we need to correctly determine $u/v = j/r$?

$$\left| \frac{y}{2^m} - \frac{j}{r} \right| \leq \frac{1}{2N^2} \quad \Rightarrow \quad \frac{u}{v} = \frac{j}{r}$$

Choosing $m = 2\log(N) + 1$ for phase estimation makes such an approximation likely.
We might get unlucky: j could have common factors with r.

A random eigenvector

$$|\psi_j\rangle = \frac{|1\rangle + \omega_r^{-j}|a\rangle + \cdots + \omega_r^{-j(r-1)}|a^{r-1}\rangle}{\sqrt{r}}$$

$$\mathcal{M}_a|\psi_j\rangle = \omega_r^j|\psi_j\rangle = e^{2\pi i \frac{j}{r}}|\psi_j\rangle$$

Suppose we're given $|\psi_j\rangle$ as a quantum state for a ***random choice*** of $j \in \{0, \ldots, r-1\}$. We can attempt to learn j/r as follows:

1. Perform phase estimation on the state $|\psi_j\rangle$ and a quantum circuit implementing $\mathcal{M}_a$. The outcome is an approximation $y/2^m \approx j/r$.
2. Among the fractions u/v in lowest terms satisfying $u, v \in \{0, \ldots, N-1\}$ and $v \neq 0$, output the one closest to $y/2^m$. This can be done efficiently using the ***continued fraction algorithm.***

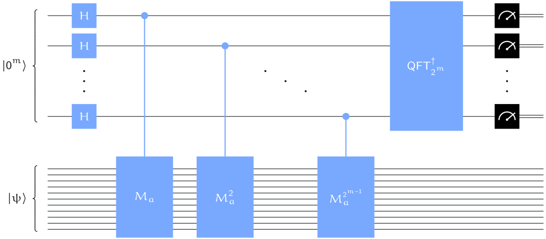
How much precision do we need to correctly determine $u/v = j/r$?

$$\left|\frac{y}{2^m} - \frac{j}{r}\right| \leq \frac{1}{2N^2} \quad \Rightarrow \quad \frac{u}{v} = \frac{j}{r}$$

If we can draw ***independent samples***, for $j \in \{0, \ldots, r-1\}$ is chosen uniformly, we can recover r with high probability by computing the ***least common multiple*** of the values of v we observed.

Implementation

To find the order of $a \in \mathbb{Z}_N^*$, we apply phase estimation to the operation $\mathcal{M}_a$. Let's measure the cost as a function of $n = \lg(N)$.



Cost for each controlled unitary

We can implement $\mathcal{M}_a$ at cost $O(n^2)$.

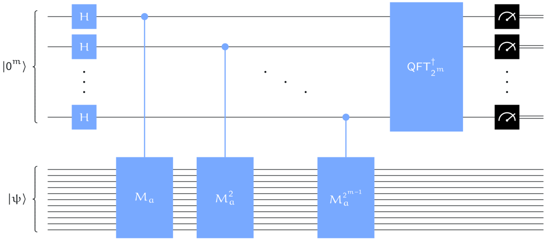
We need to implement $\mathcal{M}_a^k$ for each $k = 1, 2, 4, 8, \ldots, 2^{m-1}$. Each $\mathcal{M}_a^k$ can be implemented as follows:

- Compute $b = a^k \mod N$.
- Use a circuit for $\mathcal{M}_b$.

The cost to implement $\mathcal{M}_b = \mathcal{M}_a^k$ is $O(n^2)$.

Implementation

To find the order of $a \in \mathbb{Z}_N^*$, we apply phase estimation to the operation $\mathcal{M}_a$. Let's measure the cost as a function of $n = \lg(N)$.

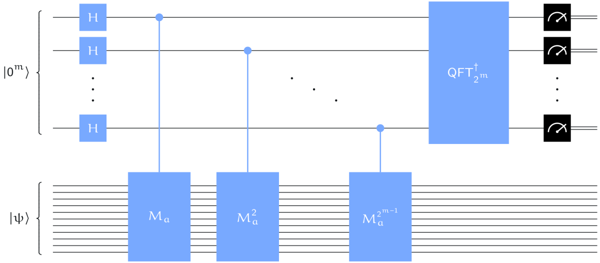


Cost for phase estimation

- m Hadamard gates: cost $O(n)$
- m controlled unitary operations: cost $O(n^3)$
- Quantum Fourier transform: cost $O(n^2)$

Total cost: $O(n^3)$

Implementation



Remaining issue: getting one of the eigenvectors $|\psi_0\rangle, \dots, |\psi_{r-1}\rangle$.
Solution: replace the eigenvector $|\psi\rangle$ with the state $|1\rangle$.
This works because of the following equation:

$$|1\rangle = \frac{|\psi_0\rangle + \cdots + |\psi_{r-1}\rangle}{\sqrt{r}}$$

The outcome is the same as if we chose $j \in \{0, 1, \dots, r-1\}$ uniformly and used $|\psi\rangle = |\psi_j\rangle$.

Factoring through order-finding

The following method succeeds in finding a factor of N with probability at least $1/2$, provided N is odd and not a prime power.

Factor-finding method

1. Choose $a \in \{2, \ldots, N-1\}$ at random.
2. Compute $d = \gcd(a, N)$. If $d \geq 2$, then output d and stop.
3. ***Compute the order*** r of a modulo N.
4. If r is even, then compute $d = \gcd(a^{r/2} - 1, N)$. If $d \geq 2$, output d and stop.
5. If this step is reached, the method has failed.

Main idea

1. By the definition of the order, we know that

$$a^r \equiv 1 \pmod{N} \iff N \text{ divides } a^r - 1$$

2. If r is even, then

$$a^r - 1 = (a^{r/2} + 1)(a^{r/2} - 1)$$

Each prime dividing N must therefore divide either $(a^{r/2} + 1)$ or $(a^{r/2} - 1)$.
For a random a, at least one of the prime factors of N is likely to divide $(a^{r/2} - 1)$.

Table of Contents

Thank you for your attention

Thanks to the IBM Quantum for great learning materials.