Introduction to Quantum Computing

Lecture slides for the Isogeny-based Cryptography School 2021

Changpeng Shao

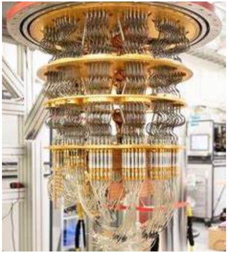
School of Mathematics, University of Bristol, UK

30th August 2021

Background

Quantum computers build on the principles of quantum mechanics. It can solve some problems much faster than the traditional computers.

A famous example is the Shor's integer factorization algorithm.

The widely used cryptosystem, RSA, relies on factoring being impossible for large integers. But Shor's algorithm shows that this problem is easy for a quantum computer.

Backgrounds

To study quantum computing, don't worry if you don't know too much about quantum mechanics. What you need to know is linear algebra in this lecture.

In this lecture, I will introduce some fundamental concepts and results. Hope to help you better understand other lectures this week.

I will not introduce the definitions in a very formal way because you can find it in many textbooks. I prefer to use examples to explain the concepts.

Qubits

Qubit (Quantum bit): $\alpha|0\rangle + \beta|1\rangle$, where $|\alpha|^2 + |\beta|^2 = 1$ and

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The special states $\{|0\rangle, |1\rangle\}$ are known as computational basis states.

Qubits

Qubit (Quantum bit): $\alpha|0\rangle + \beta|1\rangle$, where $|\alpha|^2 + |\beta|^2 = 1$ and

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The special states $\{|0\rangle, |1\rangle\}$ are known as computational basis states.

2 qubit state:

$$\alpha_{00}|0\rangle \otimes |0\rangle + \alpha_{01}|0\rangle \otimes |1\rangle + \alpha_{10}|1\rangle \otimes |0\rangle + \alpha_{11}|1\rangle \otimes |1\rangle$$

where $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$.

We will simply write $|i\rangle \otimes |j\rangle$ as $|i\rangle|j\rangle$, $|i,j\rangle$ or $|ij\rangle$.

Qubits

Qubit (Quantum bit): $\alpha|0\rangle + \beta|1\rangle$, where $|\alpha|^2 + |\beta|^2 = 1$ and

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The special states $\{|0\rangle, |1\rangle\}$ are known as computational basis states.

2 qubit state:

$$\alpha_{00}|0\rangle \otimes |0\rangle + \alpha_{01}|0\rangle \otimes |1\rangle + \alpha_{10}|1\rangle \otimes |0\rangle + \alpha_{11}|1\rangle \otimes |1\rangle$$

where $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$.

We will simply write $|i\rangle \otimes |j\rangle$ as $|i\rangle|j\rangle$, $|i, j\rangle$ or $|ij\rangle$.

n qubit state:

$$\sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

where $\sum_{x \in \{0,1\}^n} |\alpha_x|^2 = 1$.

Dirac notation or bra-ket notation

For a unit column vector $\mathbf{v} = (v_0, \ldots, v_{n-1})^T$, in quantum computing, we denote it as ("ket notation")

$$|\mathbf{v}\rangle = \sum_{j=0}^{n-1} v_j |j\rangle \leftrightarrow \begin{pmatrix} v_0 \\ \vdots \\ v_{n-1} \end{pmatrix}$$

where $\{|0\rangle, \ldots, |n-1\rangle\}$ corresponds to the standard basis of $\mathbb{C}^n$.

Its conjugate transpose is denoted as ("bra notation")

$$\langle \mathbf{v}| = \sum_{j=0}^{n-1} \bar{v}_j \langle j| \leftrightarrow \begin{pmatrix} \bar{v}_0 & \cdots & \bar{v}_{n-1} \end{pmatrix}$$

It is a row vector.

Unitary operations

Since quantum states are unit, we are allowed to use unitary operators to quantum state to keep the norm.

Unitary operations

Since quantum states are unit, we are allowed to use unitary operators to quantum state to keep the norm.

A operator U is called unitary if $UU^\dagger = U^\dagger U = I$. Here $\dagger$ is conjugate transpose.

Unitary operations

Since quantum states are unit, we are allowed to use unitary operators to quantum state to keep the norm.

A operator U is called unitary if $UU^\dagger = U^\dagger U = I$. Here $\dagger$ is conjugate transpose.

Examples:

Hadamard gate: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.

Unitary operations

Since quantum states are unit, we are allowed to use unitary operators to quantum state to keep the norm.

A operator U is called unitary if $UU^\dagger = U^\dagger U = I$. Here $\dagger$ is conjugate transpose.

Examples:

Hadamard gate: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.

Pauli matrices: $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.

Unitary operations

Since quantum states are unit, we are allowed to use unitary operators to quantum state to keep the norm.

A operator U is called unitary if $UU^\dagger = U^\dagger U = I$. Here $\dagger$ is conjugate transpose.

Examples:

Hadamard gate: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.

Pauli matrices: $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.

Control gate: $|0\rangle\langle 0| \otimes U_0 + |1\rangle\langle 1| \otimes U_1$, where U_0, U_1 are unitary operators. It means if the first qubit is $|i\rangle$, then we apply U_i to the second state.

$$\alpha_0|0\rangle|\phi_0\rangle + \alpha_1|1\rangle|\phi_1\rangle \mapsto \alpha_0|0\rangle U_0|\phi_0\rangle + \alpha_1|1\rangle U_1|\phi_1\rangle.$$

The matrix form $\begin{pmatrix} U_0 & \\ & U_1 \end{pmatrix}$.

Measurements

For a quantum state $|\phi\rangle = \sum_x \alpha_x |x\rangle$, we can measure it in the computational basis. The probability to obtain $|x\rangle$ is $|\alpha_x|^2$.

Measurements

For a quantum state $|\phi\rangle = \sum_x \alpha_x |x\rangle$, we can measure it in the computational basis. The probability to obtain $|x\rangle$ is $|\alpha_x|^2$.

For example

$$|\phi\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{\sqrt{2}}|11\rangle.$$

With probability $1/4$, we obtain $|00\rangle$, also $|01\rangle$. With probability $1/2$, we obtain $|11\rangle$.

Measurements

For a quantum state $|\phi\rangle = \sum_x \alpha_x |x\rangle$, we can measure it in the computational basis. The probability to obtain $|x\rangle$ is $|\alpha_x|^2$.

For example

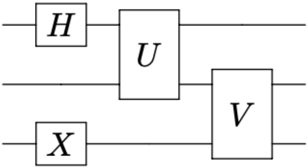
$$|\phi\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{\sqrt{2}}|11\rangle.$$

With probability $1/4$, we obtain $|00\rangle$, also $|01\rangle$. With probability $1/2$, we obtain $|11\rangle$.

We can do partial measurement. For $|\phi\rangle$, if we only measure the first qubit, then with probability $1/2$, we obtain $|0\rangle$. The state associated to $|0\rangle$ is $(|0\rangle + |1\rangle)/\sqrt{2}$.

Quantum circuit

A quantum circuit can be drawn as a diagram by associating each qubit with a horizontal "wire", and drawing each gate as a box across the wires corresponding to the qubits on which it acts.

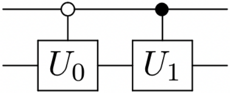


The above circuit corresponds to the unitary operator

$$(I_2 \otimes V)(U \otimes I_2)(H \otimes I_2 \otimes X)$$

on 3 qubits.

For control gate $|0\rangle\langle 0| \otimes U_0 + |1\rangle\langle 1| \otimes U_1$, the quantum circuit is



Implement classical operations in a quantum computer

Let $f : \{0,1\}^m \to \{0,1\}^n$ be a function, it becomes a unitary operator by the following trick

$$\begin{aligned} f' : \{0,1\}^m \times \{0,1\}^n &\to \{0,1\}^m \times \{0,1\}^n \\ (x, y) &\to (x, y \oplus f(x)). \end{aligned}$$

Implement classical operations in a quantum computer

Let $f : \{0,1\}^m \to \{0,1\}^n$ be a function, it becomes a unitary operator by the following trick

$$\begin{aligned} f' : \{0,1\}^m \times \{0,1\}^n &\to \{0,1\}^m \times \{0,1\}^n \\ (x, y) &\to (x, y \oplus f(x)). \end{aligned}$$

In a quantum computer, we denote it as

$$\begin{aligned} O_f : \{0,1\}^m \times \{0,1\}^n &\to \{0,1\}^m \times \{0,1\}^n \\ |x\rangle|y\rangle &\to |x\rangle|y \oplus f(x)\rangle. \end{aligned}$$

It is called an oracle to query functions.

Implement classical operations in a quantum computer

When $n = 1$, sometimes it is convenient to use

$$\begin{aligned} U_f : \{0,1\}^m &\rightarrow \{0,1\}^m \\ |x\rangle &\rightarrow (-1)^{f(x)}|x\rangle. \end{aligned}$$

Implement classical operations in a quantum computer

When $n = 1$, sometimes it is convenient to use

$$\begin{aligned} U_f : \{0,1\}^m &\rightarrow \{0,1\}^m \\ |x\rangle &\rightarrow (-1)^{f(x)}|x\rangle. \end{aligned}$$

We can implement U_f from O_f.

Implement classical operations in a quantum computer

When $n = 1$, sometimes it is convenient to use

$$\begin{aligned} U_f : \{0,1\}^m &\rightarrow \{0,1\}^m \\ |x\rangle &\rightarrow (-1)^{f(x)}|x\rangle. \end{aligned}$$

We can implement U_f from O_f.

More precisely, denote $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, then

$$|x\rangle|-\rangle \xrightarrow{O_f} \frac{1}{\sqrt{2}}|x\rangle(|f(x)\rangle - |1 \oplus f(x)\rangle).$$

If $f(x) = 0$, the result is $|x\rangle|-\rangle$; If $f(x) = 1$, the result is $-|x\rangle|-\rangle$.
In summary, the result is

$$(-1)^{f(x)}|x\rangle|-\rangle.$$

Universial set

In principle, any unitary operator on n qubits can be implemented using only 1- and 2-qubit gates. Most unitary operators on n qubits can only be realized using an exponentially large circuit of 1- and 2-qubit gates.

Universal set

In principle, any unitary operator on n qubits can be implemented using only 1- and 2-qubit gates. Most unitary operators on n qubits can only be realized using an exponentially large circuit of 1- and 2-qubit gates.

In general, we are content to give circuits that give good approximations of our desired unitary operators.

Universial set

In principle, any unitary operator on n qubits can be implemented using only 1- and 2-qubit gates. Most unitary operators on n qubits can only be realized using an exponentially large circuit of 1- and 2-qubit gates.

In general, we are content to give circuits that give good approximations of our desired unitary operators.

A set of quantum gates is called universal if any unitary operator can be approximately represented as a circuit the gates in the set.

Universial set

In principle, any unitary operator on n qubits can be implemented using only 1- and 2-qubit gates. Most unitary operators on n qubits can only be realized using an exponentially large circuit of 1- and 2-qubit gates.

In general, we are content to give circuits that give good approximations of our desired unitary operators.

A set of quantum gates is called universal if any unitary operator can be approximately represented as a circuit the gates in the set.

For example, the set $\{H, T, C\}$ with

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{-\pi i/4} \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Complexity

Gate complexity: The number of elementary gates used in the universal set.

Up to some poly-log terms, the gate complexity does not change if universal set varies.

Query complexity: The number of evaluations to the given function, i.e., the number of O_f (or U_f) used in the quantum circuit.

Deutsch-Jozsa problem

The Deutsch-Jozsa algorithm was the first to show a separation between the quantum and classical difficulty of a problem.

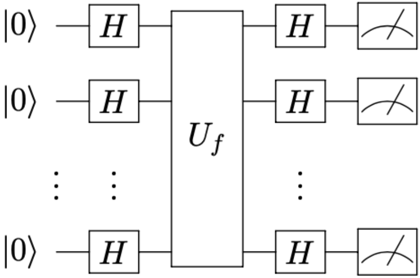
Definition 1 (Deutsch-Jozsa problem)

Let $f : \{0,1\}^n \to \{0,1\}$. It is promised to be constant or balanced (i.e., $|f^{-1}(0)| = |f^{-1}(1)| = 2^{n-1}$). The goal is to decide which is the case by making as few function evaluations as possible.

Classically, it requires $2^{n-1} + 1$ function evaluations. However, the Deutsch-Jozsa algorithm only uses 1 function evaluation.

Deutsch-Jozsa algorithm

The circuit of Deutsch-Jozsa algorithm is very simple:



The last step means measurement.

Deutsch-Jozsa algorithm

1. The initial state is $|0\rangle^{\otimes n}$.

Deutsch-Jozsa algorithm

1. The initial state is $|0\rangle^{\otimes n}$.
2. In the first step, we apply $H^{\otimes n}$, then we obtain

$$\frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle.$$

Deutsch-Jozsa algorithm

1. The initial state is $|0\rangle^{\otimes n}$.
2. In the first step, we apply $H^{\otimes n}$, then we obtain

$$\frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle.$$

3. In the second step, we apply U_f which gives

$$\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{f(y)} |y\rangle.$$

Deutsch-Jozsa algorithm

1. The initial state is $|0\rangle^{\otimes n}$.
2. In the first step, we apply $H^{\otimes n}$, then we obtain

$$\frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle.$$

3. In the second step, we apply U_f which gives

$$\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{f(y)} |y\rangle.$$

4. Finally, we apply $H^{\otimes n}$ again

$$\frac{1}{2^n} \sum_{z \in \{0,1\}^n} \left(\sum_{y \in \{0,1\}^n} (-1)^{f(y) + y \cdot z} \right) |z\rangle.$$

Deutsch-Jozsa algorithm

Now let us check the possible outputs:

Deutsch-Jozsa algorithm

Now let us check the possible outputs:

- If f is constant, say $f(y) = 0$ for all y, then

$$\frac{1}{2^n} \sum_{y \in \{0,1\}^n} (-1)^{f(y)+y\cdot z} = \begin{cases} 1 & z = 0 \\ 0 & z \neq 0 \end{cases}$$

  So final state is $|0\rangle^{\otimes n}$. If we perform measurement, we always obtain $|0\rangle^{\otimes n}$.

Deutsch-Jozsa algorithm

Now let us check the possible outputs:

- If f is constant, say $f(y) = 0$ for all y, then

$$\frac{1}{2^n} \sum_{y \in \{0,1\}^n} (-1)^{f(y) + y \cdot z} = \begin{cases} 1 & z = 0 \\ 0 & z \neq 0 \end{cases}$$

  So final state is $|0\rangle^{\otimes n}$. If we perform measurement, we always obtain $|0\rangle^{\otimes n}$.

- If f is balanced, then the coefficient of $|0\rangle^{\otimes n}$

$$\frac{1}{2^n} \sum_{y \in \{0,1\}^n} (-1)^{f(y)} = 0.$$

  We therefore never obtain $|0\rangle^{\otimes n}$ by measuring the final state.

Simon's problem

Simon's algorithm was the first quantum algorithm to show an exponential speed-up versus the best classical algorithm.

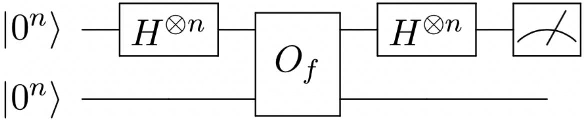
Definition 2 (Simon's problem)

Let $f : \{0,1\}^n \to \{0,1\}^n$. There is a unknown s such that $f(x) = f(y)$ if and only if $y = x \oplus s$. The goal is to find s.

The classical algorithm needs at least $2^{n/2}$ queries to f. While Simon's algorithm only uses $O(n)$ queries.

Simon's algorithm

The circuit of Simon's algorithm is very similar to the circuit of Deutsch-Jozsa algorithm:



Simon's algorithm

1. The initial state is $|0^n\rangle|0^n\rangle$.

Simon's algorithm

1. The initial state is $|0^n\rangle|0^n\rangle$.
2. In the first step, we apply $H^{\otimes n} \otimes I$

$$\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle|0^n\rangle.$$

Simon's algorithm

1. The initial state is $|0^n\rangle|0^n\rangle$.
2. In the first step, we apply $H^{\otimes n} \otimes I$

$$\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle|0^n\rangle.$$

3. In the second step, we apply O_f

$$\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle|f(y)\rangle.$$

Simon's algorithm

1. The initial state is $|0^n\rangle|0^n\rangle$.
2. In the first step, we apply $H^{\otimes n} \otimes I$

$$\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle|0^n\rangle.$$

3. In the second step, we apply O_f

$$\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle|f(y)\rangle.$$

4. Finally we apply $H^{\otimes n} \otimes I$ again

$$\frac{1}{2^n} \sum_{z \in \{0,1\}^n} \sum_{y \in \{0,1\}^n} (-1)^{y \cdot z}|z\rangle|f(y)\rangle.$$

Simon's algorithm

Recall that $f(x) = f(y)$ iff $y = x \oplus s$, so we can split $\{0,1\}^n$ into $A \cup (A \oplus s)$. On A, f is one-to-one.

Simon's algorithm

Recall that $f(x) = f(y)$ iff $y = x \oplus s$, so we can split $\{0,1\}^n$ into $A \cup (A \oplus s)$. On A, f is one-to-one.

In the final state,

$$\begin{aligned}
\sum_{y \in \{0,1\}^n} (-1)^{y \cdot z} |z\rangle &= \sum_{y \in A} \left((-1)^{y \cdot z} + (-1)^{(y \oplus s) \cdot z} \right) |z\rangle \\
&= \sum_{y \in A} (-1)^{y \cdot z} \left(1 + (-1)^{s \cdot z} \right) |z\rangle .
\end{aligned}$$

The coefficient is nonzero if $s \cdot z = 0$.

Simon's algorithm

Recall that $f(x) = f(y)$ iff $y = x \oplus s$, so we can split $\{0,1\}^n$ into $A \cup (A \oplus s)$. On A, f is one-to-one.

In the final state,

$$
\begin{aligned}
\sum_{y \in \{0,1\}^n} (-1)^{y \cdot z} |z\rangle &= \sum_{y \in A} \left((-1)^{y \cdot z} + (-1)^{(y \oplus s) \cdot z} \right) |z\rangle \\
&= \sum_{y \in A} (-1)^{y \cdot z} \left(1 + (-1)^{s \cdot z} \right) |z\rangle.
\end{aligned}
$$

The coefficient is nonzero if $s \cdot z = 0$.

If we run the above process $n-1$ times, we obtain $z_1, \ldots, z_{n-1}$ such that $s \cdot z_i = 0$ for all i. From this linear system, we can determine s.

Quantum Fourier Transform (QFT)

Definition 3 (Quantum Fourier Transform (QFT))

Let N be a integer, $\omega = e^{2\pi i/N}$, the QFT is defined by

$$\begin{aligned} Q_N : \mathbb{Z}_N &\rightarrow \mathbb{Z}_N \\ |x\rangle &\mapsto \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega^{xy} |y\rangle. \end{aligned}$$

- A very important unitary operator in quantum information theory.
- It is the normalized discrete Fourier transform.

Quantum Fourier Transform (QFT)

In matrix form:

$$Q_N = \frac{1}{\sqrt{N}} \sum_{x,y \in \mathbb{Z}_N} \omega^{xy} |y\rangle\langle x|.$$

The inverse of QFT is

$$Q_N^{-1} : |x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega^{-xy} |y\rangle.$$

Check:

$$Q_N^{-1} Q_N |x\rangle = \frac{1}{N} \sum_{z \in \mathbb{Z}_N} \left(\sum_{y \in \mathbb{Z}_N} \omega^{y(x-z)} \right) |z\rangle = |x\rangle.$$

Quantum Fourier Transform (QFT)

Example 4

$$Q_2 = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \quad Q_3 = \frac{1}{\sqrt{3}}\begin{pmatrix}1 & 1 & 1\\ 1 & e^{2\pi i/3} & e^{-2\pi i/3}\\ 1 & e^{-2\pi i/3} & e^{2\pi i/3}\end{pmatrix},$$

$$Q_4 = \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & i & -1 & -i\\ 1 & -1 & 1 & -1\\ 1 & -i & -1 & i\end{pmatrix}.$$

Efficient implementation of the QFT

It can be implemented using $O(\log^2 N)$ elementary quantum gates:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad R_d = \begin{pmatrix} 1 & 0 \\ 0 & e^{\pi i/2^d} \end{pmatrix}.$$

Efficient implementation of the QFT

It can be implemented using $O(\log^2 N)$ elementary quantum gates:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad R_d = \begin{pmatrix} 1 & 0 \\ 0 & e^{\pi i/2^d} \end{pmatrix}.$$

Let's take a look at the case $N = 8$:

$$\begin{aligned} & Q_8|x_0, x_1, x_2\rangle \\ = \quad & \frac{1}{\sqrt{8}} \sum_{y_0,y_1,y_2=0}^{1} e^{2\pi i(x(y_0+2y_1+4y_2))/8}|y_0, y_1, y_2\rangle \\ = \quad & \frac{1}{\sqrt{8}} \left(\sum_{y_0=0}^{1} e^{\pi i x y_0/4}|y_0\rangle \right) \left(\sum_{y_1=0}^{1} e^{\pi i x y_1/2}|y_1\rangle \right) \left(\sum_{y_2=0}^{1} e^{\pi i x y_2}|y_2\rangle \right) \end{aligned}$$

Note: $|x\rangle = |x_0, x_1, x_2\rangle$ and $x = x_0 + 2x_1 + 4x_2$ is the binary form.

Efficient implementation of the QFT

$$\sum_{y_0=0}^{1} e^{\pi i x y_0/4}|y_0\rangle \quad = \quad \sum_{y_0=0}^{1} e^{\pi i x_0 y_0/4} e^{\pi i x_1 y_0/2} e^{\pi i x_2 y_0}|y_0\rangle$$

Efficient implementation of the QFT

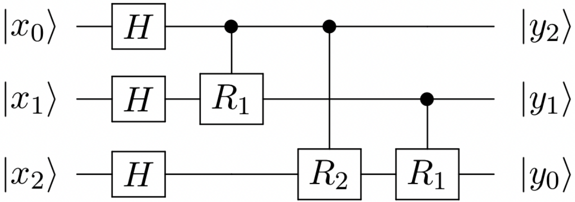
$$
\begin{aligned}
\sum_{y_0=0}^{1} e^{\pi i x y_0/4} |y_0\rangle &= \sum_{y_0=0}^{1} e^{\pi i x_0 y_0/4} e^{\pi i x_1 y_0/2} e^{\pi i x_2 y_0} |y_0\rangle \\
\sum_{y_1=0}^{1} e^{\pi i x y_1/2} |y_1\rangle &= \sum_{y_1=0}^{1} e^{\pi i x_0 y_1/2} e^{\pi i x_1 y_1} |y_1\rangle
\end{aligned}
$$

Efficient implementation of the QFT

$$
\begin{aligned}
\sum_{y_0=0}^{1} e^{\pi i x y_0/4} |y_0\rangle &= \sum_{y_0=0}^{1} e^{\pi i x_0 y_0/4} e^{\pi i x_1 y_0/2} e^{\pi i x_2 y_0} |y_0\rangle \\
\sum_{y_1=0}^{1} e^{\pi i x y_1/2} |y_1\rangle &= \sum_{y_1=0}^{1} e^{\pi i x_0 y_1/2} e^{\pi i x_1 y_1} |y_1\rangle \\
\sum_{y_2=0}^{1} e^{\pi i x y_2} |y_2\rangle &= \sum_{y_2=0}^{1} e^{\pi i x_0 y_2} |y_2\rangle.
\end{aligned}
$$

Efficient implementation of the QFT

$$
\begin{aligned}
\sum_{y_0=0}^{1} e^{\pi i x y_0/4}|y_0\rangle &= \sum_{y_0=0}^{1} e^{\pi i x_0 y_0/4} e^{\pi i x_1 y_0/2} e^{\pi i x_2 y_0}|y_0\rangle \\
\sum_{y_1=0}^{1} e^{\pi i x y_1/2}|y_1\rangle &= \sum_{y_1=0}^{1} e^{\pi i x_0 y_1/2} e^{\pi i x_1 y_1}|y_1\rangle \\
\sum_{y_2=0}^{1} e^{\pi i x y_2}|y_2\rangle &= \sum_{y_2=0}^{1} e^{\pi i x_0 y_2}|y_2\rangle.
\end{aligned}
$$



Applications of QFT: quantum phase estimation (QPE)

An important subroutine of many quantum algorithms.

Applications of QFT: quantum phase estimation (QPE)

An important subroutine of many quantum algorithms.

Input: a unitary operator U and an eigenvector $|\psi\rangle$.

Output: $\theta \in [0, 2\pi)$ such that $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$.

Applications of QFT: quantum phase estimation (QPE)

An important subroutine of many quantum algorithms.

Input: a unitary operator U and an eigenvector $|\psi\rangle$.

Output: $\theta \in [0, 2\pi)$ such that $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$.

1. Prepare the initial state $|0^n\rangle|\psi\rangle$.

Applications of QFT: quantum phase estimation (QPE)

An important subroutine of many quantum algorithms.

Input: a unitary operator U and an eigenvector $|\psi\rangle$.

Output: $\theta \in [0, 2\pi)$ such that $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$.

1. Prepare the initial state $|0^n\rangle|\psi\rangle$.
2. Apply Hadamard gates $H^{\otimes n}$ to the first register:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|\psi\rangle.$$

Applications of QFT: quantum phase estimation (QPE)

An important subroutine of many quantum algorithms.

Input: a unitary operator U and an eigenvector $|\psi\rangle$.

Output: $\theta \in [0, 2\pi)$ such that $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$.

1. Prepare the initial state $|0^n\rangle|\psi\rangle$.
2. Apply Hadamard gates $H^{\otimes n}$ to the first register:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|\psi\rangle.$$

3. Apply control gate $\sum_x |x\rangle\langle x| \otimes U^x$:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i x\theta}|x\rangle|\psi\rangle.$$

Applications of QFT: quantum phase estimation (QPE)

An important subroutine of many quantum algorithms.

Input: a unitary operator U and an eigenvector $|\psi\rangle$.

Output: $\theta \in [0, 2\pi)$ such that $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$.

1. Prepare the initial state $|0^n\rangle|\psi\rangle$.
2. Apply Hadamard gates $H^{\otimes n}$ to the first register:
$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|\psi\rangle.$$
3. Apply control gate $\sum_x |x\rangle\langle x| \otimes U^x$:
$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i x\theta}|x\rangle|\psi\rangle.$$
4. Apply QFT^{-1} to $|x\rangle$:
$$\frac{1}{2^n} \sum_{y=0}^{2^n-1} \left(\sum_{x=0}^{2^n-1} e^{2\pi i x(\theta - y/2^n)} \right) |y\rangle|\psi\rangle.$$

Applications of QFT: quantum phase estimation (QPE)

Denote $\delta_y = \theta - y/2^n$. The coefficient of $|y\rangle|\psi\rangle$ is

$$\frac{1}{2^n}\left|\sum_{x=0}^{2^n-1} e^{2\pi i \delta_y x}\right| = \frac{1}{2^n}\left|\frac{e^{2\pi i \delta_y 2^n}-1}{e^{2\pi i \delta_y}-1}\right| = \frac{1}{2^n}\left|\frac{\sin(\pi\delta_y 2^n)}{\sin(\pi\delta_y)}\right|.$$

Applications of QFT: quantum phase estimation (QPE)

Denote $\delta_y = \theta - y/2^n$. The coefficient of $|y\rangle|\psi\rangle$ is

$$\frac{1}{2^n}\left|\sum_{x=0}^{2^n-1} e^{2\pi i \delta_y x}\right| = \frac{1}{2^n}\left|\frac{e^{2\pi i \delta_y 2^n}-1}{e^{2\pi i \delta_y}-1}\right| = \frac{1}{2^n}\left|\frac{\sin(\pi\delta_y 2^n)}{\sin(\pi\delta_y)}\right|.$$

If $|\delta_y|2^n \leq 1/2$, then the above quantity is lower bounded by

$$\geq \frac{1}{2^n}\left|\frac{2\delta_y 2^n}{\pi\delta_y}\right| = \frac{2}{\pi}$$

based on the fact $\sin(t) \geq 2t/\pi$ when $|t| \leq \pi/2$.

Applications of QFT: quantum phase estimation (QPE)

Denote $\delta_y = \theta - y/2^n$. The coefficient of $|y\rangle|\psi\rangle$ is

$$\frac{1}{2^n}\left|\sum_{x=0}^{2^n-1} e^{2\pi i \delta_y x}\right| = \frac{1}{2^n}\left|\frac{e^{2\pi i \delta_y 2^n}-1}{e^{2\pi i \delta_y}-1}\right| = \frac{1}{2^n}\left|\frac{\sin(\pi\delta_y 2^n)}{\sin(\pi\delta_y)}\right|.$$

If $|\delta_y|2^n \leq 1/2$, then the above quantity is lower bounded by

$$\geq \frac{1}{2^n}\left|\frac{2\delta_y 2^n}{\pi\delta_y}\right| = \frac{2}{\pi}$$

based on the fact $\sin(t) \geq 2t/\pi$ when $|t| \leq \pi/2$.

This means by measurement, we obtain y such that $y/2^n \approx \theta$.
The success probability is at least $4/\pi^2$.

Applications of QFT: quantum phase estimation (QPE)

Denote $\delta_y = \theta - y/2^n$. The coefficient of $|y\rangle|\psi\rangle$ is

$$\frac{1}{2^n}\left|\sum_{x=0}^{2^n-1} e^{2\pi i \delta_y x}\right| = \frac{1}{2^n}\left|\frac{e^{2\pi i \delta_y 2^n}-1}{e^{2\pi i \delta_y}-1}\right| = \frac{1}{2^n}\left|\frac{\sin(\pi\delta_y 2^n)}{\sin(\pi\delta_y)}\right|.$$

If $|\delta_y|2^n \leq 1/2$, then the above quantity is lower bounded by

$$\geq \frac{1}{2^n}\left|\frac{2\delta_y 2^n}{\pi\delta_y}\right| = \frac{2}{\pi}$$

based on the fact $\sin(t) \geq 2t/\pi$ when $|t| \leq \pi/2$.

This means by measurement, we obtain y such that $y/2^n \approx \theta$. The success probability is at least $4/\pi^2$.

We can modify the algorithm to ensure the success probability is at least $1-\epsilon$ for arbitrary small ϵ.

Applications of QFT: period finding

One of the most important applications of the QFT, the key step of Shor's algorithm.

Applications of QFT: period finding

One of the most important applications of the QFT, the key step of Shor's algorithm.

Imagine we are given access to an oracle O_f function $f : \mathbb{Z}_N \to \mathbb{Z}_M$, for some integers N and M, such that:

- f is periodic: there exists r such that r divides N and $f(x+r) = f(x)$ for all $x \in \mathbb{Z}_N$;
- f is one-to-one on each period: for all pairs (x, y) such that $|x - y| < r$, $f(x) \neq f(y)$.

Our task is to determine r.

Recall: $O_f : |x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$.

Applications of QFT: period finding

- We start with the state $|0..0\rangle|0..0\rangle$.

Applications of QFT: period finding

- We start with the state $|0..0\rangle|0..0\rangle$.
- Apply Hadamard gates to the first register:

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|0^m\rangle.$$

Applications of QFT: period finding

- We start with the state $|0..0\rangle|0..0\rangle$.
- Apply Hadamard gates to the first register:

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|0^m\rangle.$$

- Apply O_f to the two registers:

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle = \frac{1}{\sqrt{N}}\sum_{y=0}^{r-1}\left(\sum_{j=0}^{N/r-1}|y+jr\rangle\right)|f(y)\rangle$$

Applications of QFT: period finding

- We start with the state $|0..0\rangle|0..0\rangle$.
- Apply Hadamard gates to the first register:

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|0^m\rangle.$$

- Apply O_f to the two registers:

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle = \frac{1}{\sqrt{N}}\sum_{y=0}^{r-1}\left(\sum_{j=0}^{N/r-1}|y+jr\rangle\right)|f(y)\rangle$$

- Measure the second register: obtain a random y

$$\frac{\sqrt{r}}{\sqrt{N}}\sum_{j=0}^{N/r-1}|y+jr\rangle$$

Applications of QFT: period finding

- Apply Q_N to the first register: $\omega = e^{2\pi i/N}$

$$\frac{\sqrt{r}}{N}\sum_{z=0}^{N-1}\omega^{yz}\left(\sum_{j=0}^{N/r-1}\omega^{jrz}\right)|z\rangle.$$

Applications of QFT: period finding

- Apply Q_N to the first register: $\omega = e^{2\pi i/N}$

$$\frac{\sqrt{r}}{N}\sum_{z=0}^{N-1}\omega^{yz}\left(\sum_{j=0}^{N/r-1}\omega^{jrz}\right)|z\rangle.$$

Note that if $\omega^{rz} \neq 1$, i.e., $rz \not\equiv 0 \mod N$, then

$$\sum_{j=0}^{N/r-1}\omega^{jrz} = \frac{\omega^{rzN}-1}{\omega^{rz}-1} = 0.$$

So the state is

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|\frac{N}{r}s\rangle.$$

Applications of QFT: period finding

- Apply Q_N to the first register: $\omega = e^{2\pi i/N}$

$$\frac{\sqrt{r}}{N}\sum_{z=0}^{N-1}\omega^{yz}\left(\sum_{j=0}^{N/r-1}\omega^{jrz}\right)|z\rangle.$$

Note that if $\omega^{rz} \neq 1$, i.e., $rz \not\equiv 0 \mod N$, then

$$\sum_{j=0}^{N/r-1}\omega^{jrz} = \frac{\omega^{rzN}-1}{\omega^{rz}-1} = 0.$$

So the state is

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|\frac{N}{r}s\rangle.$$

Measure it we obtain a random s (unknown) and z (known) such that $z/N = s/r$. If s is coprime to r, then we can determine r by simplify z/N. This happens with probability at least $1/\log\log r$.

Shor's algorithm

Input: Integer N
Output: integers p, q such that $N = pq$

[1] gcd = greatest common divisor.
[2] order: the minimal $r > 0$ s.t. $a^r \equiv 1 \mod N$.

Shor's algorithm

Input: Integer N
Output: integers p, q such that $N = pq$

1. Choose $1 < a < N$ uniformly at random.

[1] gcd = greatest common divisor.
[2] order: the minimal $r > 0$ s.t. $a^r \equiv 1 \mod N$.

Shor's algorithm

Input: Integer N

Output: integers p, q such that $N = pq$

1. Choose $1 < a < N$ uniformly at random.
2. Compute $b = \gcd(a, N)$. If $b > 1$ output b and stop. [1]

[1] gcd = greatest common divisor.

[2] order: the minimal $r > 0$ s.t. $a^r \equiv 1 \mod N$.

Shor's algorithm

Input: Integer N

Output: integers p, q such that $N = pq$

1. Choose $1 < a < N$ uniformly at random.
2. Compute $b = \gcd(a, N)$. If $b > 1$ output b and stop. [1]
3. Compute the order r of a. If r is odd, go to step 1. [2]

[1] gcd = greatest common divisor.

[2] order: the minimal $r > 0$ s.t. $a^r \equiv 1 \mod N$.

Shor's algorithm

Input: Integer N

Output: integers p, q such that $N = pq$

1. Choose $1 < a < N$ uniformly at random.
2. Compute $b = \gcd(a, N)$. If $b > 1$ output b and stop. [1]
3. Compute the order r of a. If r is odd, go to step 1. [2]
4. Compute $s = \gcd(a^{r/2} - 1, N)$. If $s = 1$, go to step 1.

[1] gcd = greatest common divisor.

[2] order: the minimal $r > 0$ s.t. $a^r \equiv 1 \mod N$.

Shor's algorithm

Input: Integer N

Output: integers p, q such that $N = pq$

1. Choose $1 < a < N$ uniformly at random.
2. Compute $b = \gcd(a, N)$. If $b > 1$ output b and stop. [1]
3. Compute the order r of a. If r is odd, go to step 1. [2]
4. Compute $s = \gcd(a^{r/2} - 1, N)$. If $s = 1$, go to step 1.
5. Output $s, N/s$.

[1] gcd = greatest common divisor.

[2] order: the minimal $r > 0$ s.t. $a^r \equiv 1 \mod N$.

Shor's algorithm

Input: Integer N

Output: integers p, q such that $N = pq$

1. Choose $1 < a < N$ uniformly at random.
2. Compute $b = \gcd(a, N)$. If $b > 1$ output b and stop. [1]
3. Compute the order r of a. If r is odd, go to step 1. [2]
4. Compute $s = \gcd(a^{r/2} - 1, N)$. If $s = 1$, go to step 1.
5. Output $s, N/s$.

Step 3 is technical, it relates to period finding. Consider

$$f(x) = a^x \mod N.$$

We can check that f is periodic with period r and one-to-one on each period.

[1] gcd = greatest common divisor.

[2] order: the minimal $r > 0$ s.t. $a^r \equiv 1 \mod N$.

Grover's algorithm

A simple example of a problem that fits into the query complexity model is the unstructured search problem.

Definition 5 (Grover's search problem)

Given access to a function $f : \mathbb{Z}_N \to \{0,1\}$ with the promise that $f(x_0) = 1$ for a unique element x_0. Our task is to determine x_0.

Classical algorithm: N queries (i.e., N function evaluations to f).
Quantum algorithm: $O(\sqrt{N})$ queries.

Grover's algorithm

1. Prepare $|\phi\rangle = H^{\otimes n}|0^n\rangle$

Grover's algorithm

1. Prepare $|\phi\rangle = H^{\otimes n}|0^n\rangle$
2. Repeat the following operations $O(\sqrt{N})$ times:
   2.1 Apply U_f
   2.2 Apply $D := -H^{\otimes n}U_0H^{\otimes n}$, where U_0 maps $|0^n\rangle$ to $-|0^n\rangle$ and keeps all other basis states invariant.

Grover's algorithm

1. Prepare $|\phi\rangle = H^{\otimes n}|0^n\rangle$
2. Repeat the following operations $O(\sqrt{N})$ times:
   - 2.1 Apply U_f
   - 2.2 Apply $D := -H^{\otimes n}U_0 H^{\otimes n}$, where U_0 maps $|0^n\rangle$ to $-|0^n\rangle$ and keeps all other basis states invariant.
3. Measure all the qubits and output the result.

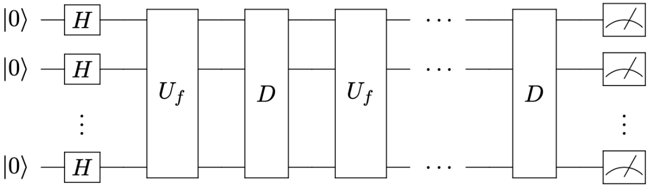
Recall: $U_f|x\rangle = (-1)^{f(x)}|x\rangle$. This is a reflection.

$D = -H^{\otimes n}(I - 2|0^n\rangle\langle 0^n|)H^{\otimes n} = 2H^{\otimes n}|0^n\rangle\langle 0^n|H^{\otimes n} - I$ is another reflection.

So DU_f is a rotation (need some analysis).

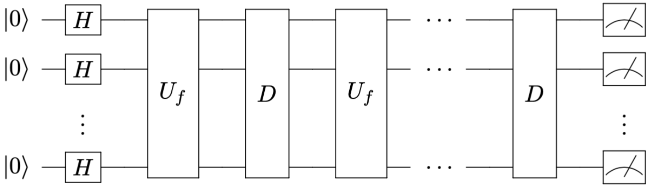
Grover's algorithm

In circuit diagram form, Grover's algorithm looks like this:



Grover's algorithm

In circuit diagram form, Grover's algorithm looks like this:



Note that

$$|\phi\rangle = H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle.$$

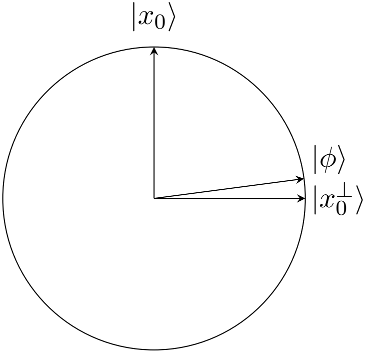
It formally equals

$$|\phi\rangle = \frac{1}{\sqrt{N}}|x_0\rangle + \frac{\sqrt{N-1}}{\sqrt{N}}|x_0^{\perp}\rangle,$$

where

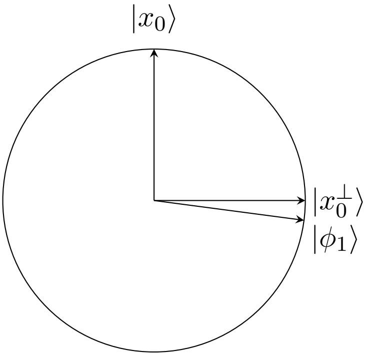
$$|x_0^{\perp}\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \in \mathbb{Z}_N, x \neq x_0} |x\rangle.$$

Grover's algorithm: Geometric argument



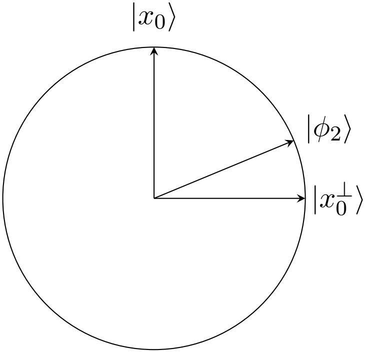
$$|\phi\rangle = \sin\theta|x_0\rangle + \cos\theta|x_0^\perp\rangle$$

Grover's algorithm: Geometric argument



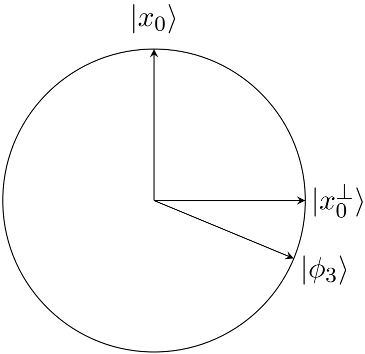
$$|\phi_1\rangle = U_f|\phi\rangle$$

Grover's algorithm: Geometric argument



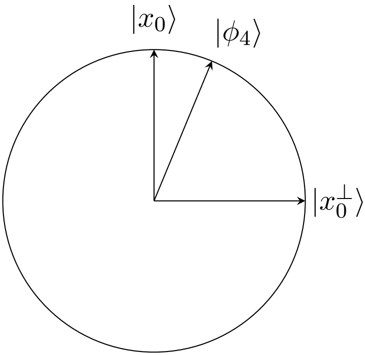
$$|\phi_2\rangle = D|\phi_1\rangle$$

Grover's algorithm: Geometric argument



$$|\phi_3\rangle = U_f|\phi_2\rangle$$

Grover's algorithm: Geometric argument



$$|\phi_4\rangle = D|\phi_3\rangle$$

Grover's algorithm

We denote $\sin\theta = 1/\sqrt{N}$ and $\cos\theta = \sqrt{N-1}/\sqrt{N}$. In step 3, we can denote $U_0 = I - 2|0^n\rangle\langle 0^n|$ so that $D = -(I - 2|\phi\rangle\langle\phi|)$.

Grover's algorithm

We denote $\sin\theta = 1/\sqrt{N}$ and $\cos\theta = \sqrt{N-1}/\sqrt{N}$. In step 3, we can denote $U_0 = I - 2|0^n\rangle\langle 0^n|$ so that $D = -(I - 2|\phi\rangle\langle\phi|)$.
So in step 3, first U_f maps $|\phi\rangle$ to

$$-\sin\theta|x_0\rangle + \cos\theta|x_0^{\perp}\rangle.$$

Grover's algorithm

We denote $\sin\theta = 1/\sqrt{N}$ and $\cos\theta = \sqrt{N-1}/\sqrt{N}$. In step 3, we can denote $U_0 = I - 2|0^n\rangle\langle 0^n|$ so that $D = -(I - 2|\phi\rangle\langle\phi|)$.
So in step 3, first U_f maps $|\phi\rangle$ to

$$-\sin\theta|x_0\rangle + \cos\theta|x_0^\perp\rangle.$$

Then apply D to obtain

$$\begin{aligned}
& \sin\theta(I - 2|\phi\rangle\langle\phi|)|x_0\rangle - \cos\theta(I - 2|\phi\rangle\langle\phi|)|x_0^\perp\rangle \\
= & \sin\theta(|x_0\rangle - 2\sin\theta(\sin\theta|x_0\rangle + \cos\theta|x_0^\perp\rangle)) \\
& - \cos\theta(|x_0^\perp\rangle - 2\cos\theta(\sin\theta|x_0\rangle + \cos\theta|x_0^\perp\rangle)) \\
= & \sin(3\theta)|x_0\rangle + \cos(3\theta)|x_0^\perp\rangle.
\end{aligned}$$

Grover's algorithm

We denote $\sin\theta = 1/\sqrt{N}$ and $\cos\theta = \sqrt{N-1}/\sqrt{N}$. In step 3, we can denote $U_0 = I - 2|0^n\rangle\langle 0^n|$ so that $D = -(I - 2|\phi\rangle\langle\phi|)$.
So in step 3, first U_f maps $|\phi\rangle$ to

$$-\sin\theta|x_0\rangle + \cos\theta|x_0^\perp\rangle.$$

Then apply D to obtain

$$\begin{aligned}
& \sin\theta(I - 2|\phi\rangle\langle\phi|)|x_0\rangle - \cos\theta(I - 2|\phi\rangle\langle\phi|)|x_0^\perp\rangle \\
= & \sin\theta(|x_0\rangle - 2\sin\theta(\sin\theta|x_0\rangle + \cos\theta|x_0^\perp\rangle)) \\
& - \cos\theta(|x_0^\perp\rangle - 2\cos\theta(\sin\theta|x_0\rangle + \cos\theta|x_0^\perp\rangle)) \\
= & \sin(3\theta)|x_0\rangle + \cos(3\theta)|x_0^\perp\rangle.
\end{aligned}$$

As we have seen, DU_f is the product of two reflections in the plane spanned by $\{|x_0\rangle, |x_0^\perp\rangle\}$. So DU_f is a rotation of angle 2θ.

Grover's algorithm

Hence, after T steps of iteration we obtain

$$\sin((2T+1)\theta)|x_0\rangle + \cos((2T+1)\theta)|x_0^{\perp}\rangle.$$

Grover's algorithm

Hence, after T steps of iteration we obtain

$$\sin((2T+1)\theta)|x_0\rangle + \cos((2T+1)\theta)|x_0^{\perp}\rangle.$$

Since $\sin\theta = 1/\sqrt{N}$, we have $\theta \approx 1/\sqrt{N}$. To make $\sin((2T+1)\theta)$ close to 1, we can choose T so that $(2T+1)\theta \approx \pi/2$. Namely, $T \approx \sqrt{N}\pi/4 - 1/2$.

Further readings

You may find the following lecture notes and books useful:

- Lecture Notes on Quantum Algorithms, Andrew Childs, University of Maryland
  http://www.cs.umd.edu/~amchilds/qa/
  An excellent resource for more advanced topics on quantum algorithms.
- Quantum Computing: Lecture Notes, Ronald de Wolf, QuSoft, CWI and University of Amsterdam
  https://export.arxiv.org/abs/1907.09415
  A comprehensive lecture note for more topics on quantum computing.
- Quantum Computation and Quantum Information, Nielsen and Chuang
  Cambridge University Press, 2001
  The Bible of quantum computing.