Qiskit Fall Fest 2025: Malaysia

Hosts: Lee Chang Xin[3], Choong Pak Shen[1,5]
Advisors: Vannajan Sanghiran Lee[1,2], Nurisya Mohd Shah[1,4,5]

[1]Malaysia Quantum Information Initiative (MyQI)
[2]CoE Quantum Information Science and Technology (QIST), Universiti Malaya
[3]Department of Physics, Universiti Malaya
[4]Department of Physics, Universiti Putra Malaysia
[5]Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia

November 7, 2025

Universiti Malaya student team

1. Mok Zhen Yang
2. Chee Tian Hou
3. Tan Kai Zhe
4. Errol Tay Lee Han
5. Tan Yee Tern
6. Tee Hui En

Universiti Putra Malaysia student team

1. Ain Nabihah Mohd Padaliah
2. Nur Ainin Sabrina Nor Efandi
3. Ricardo André González Gómez
4. Nur Amirah Hafizah Abdul Wahab
5. Nurul Aisyah Azhar

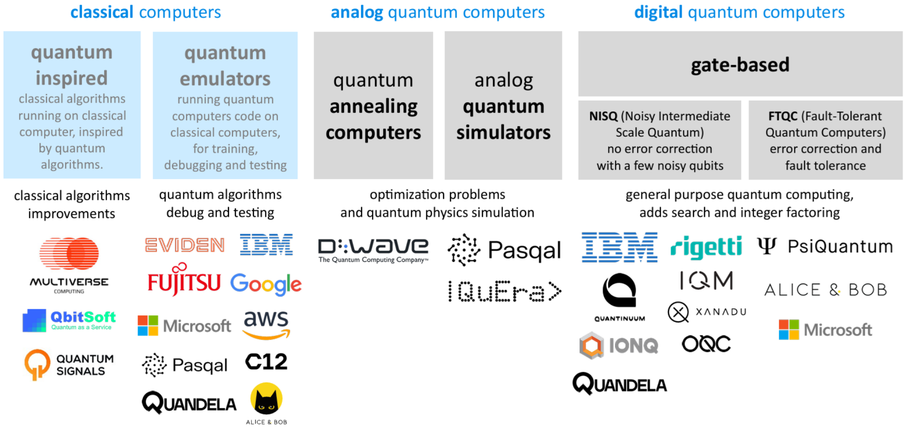
Workshop on quantum algorithms

Timetable

Day 1	**30 October 2025**
0800 - 0900	Registration
0900 - 1200	Introduction to quantum information and quantum computing
1200 - 1330	Lunch break
1330 - 1630	Deutsch-Jozsa algorithm
Day 2	**31 October 2025**
0900 - 1200	Shor's algorithm
1200 - 1500	Lunch break and prayer time
1500 - 1545	Yap Yung Szen: Control System for Superconducting Quantum Computers
1545 - 1630	Tomasz Paterek: Quantum Reservoir Processing: NISQ AI

Introduction to quantum information and computing
9.00am - 12.00pm GMT+8, 30 October 2025

Quantum computing



classical computers

quantum inspired
classical algorithms running on classical computer, inspired by quantum algorithms.

classical algorithms improvements

quantum emulators
running quantum computers code on classical computers, for training, debugging and testing

quantum algorithms debug and testing

analog quantum computers

quantum **annealing computers**

analog **quantum simulators**

optimization problems and quantum physics simulation

digital quantum computers

gate-based

NISQ (Noisy Intermediate Scale Quantum) no error correction with a few noisy qubits

FTQC (Fault-Tolerant Quantum Computers) error correction and fault tolerance

general purpose quantum computing, adds search and integer factoring

Figure 1: Different computing paradigms with quantum systems, hybrid systems and classical systems (Ezratty, 2025).

Quantum emulator

1. Classical software and hardware that can execute quantum algorithms which are designed to run on quantum computers.
2. This terminology coincides with the classical view of an emulator, which runs some software code on one machine that was designed for older hardware.

Quantum simulator

1. Quantum computing system that is used to simulate low temperature physics and many-body quantum physics, as envisioned by Richard Feynman.

Quantum-inspired algorithm

1. Classical algorithm that runs on classical hardware with new efficiencies inspired by quantum algorithm.

Quantum algorithm

1. Algorithm that runs on a realistic quantum computer and uses some essential quantum phenomena.

NISQ algorithm

1. Quantum algorithm that is designed for quantum processors in the noisy intermediate-scale quantum (NISQ) era.
2. Usually, some calculations are offloaded to classical processors.

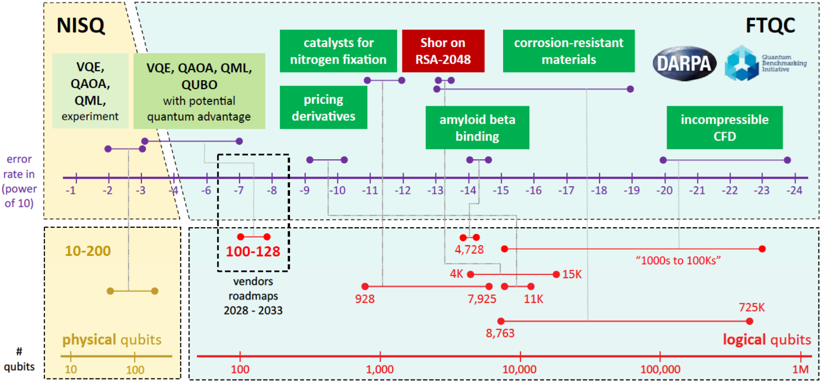
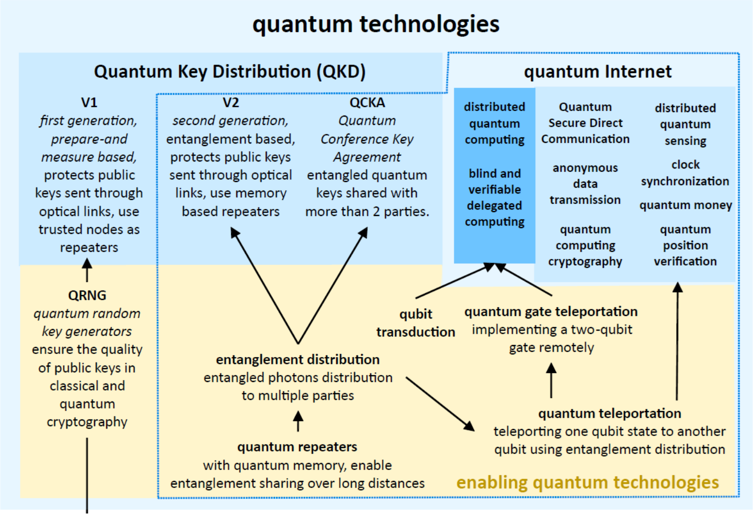


Figure 2: Algorithmic-level resource estimates for key algorithms which have some industry relevance (Ezratty, 2025).

Quantum communication



quantum technologies

Quantum Key Distribution (QKD)

V1
first generation, prepare-and measure based,
protects public keys sent through optical links, use trusted nodes as repeaters

V2
second generation,
entanglement based, protects public keys sent through optical links, use memory based repeaters

QCKA
Quantum Conference Key Agreement
entangled quantum keys shared with more than 2 parties.

quantum Internet

distributed quantum computing

blind and verifiable delegated computing

Quantum Secure Direct Communication

anonymous data transmission

quantum computing cryptography

distributed quantum sensing

clock synchronization

quantum money

quantum position verification

QRNG
quantum random key generators
ensure the quality of public keys in classical and quantum cryptography

qubit transduction

quantum gate teleportation
implementing a two-qubit gate remotely

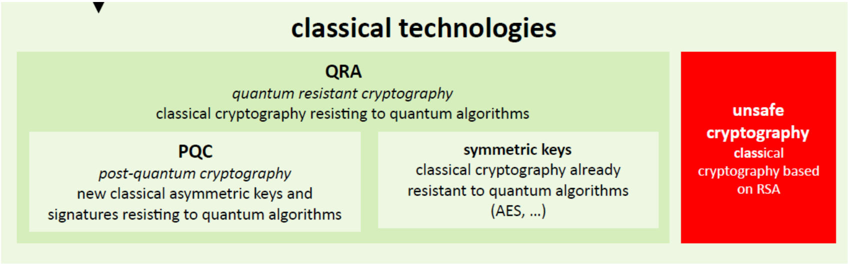
entanglement distribution
entangled photons distribution to multiple parties

quantum teleportation
teleporting one qubit state to another qubit using entanglement distribution

quantum repeaters
with quantum memory, enable entanglement sharing over long distances

enabling quantum technologies

Figure 3: Various types of quantum communication and cybersecurity technologies (quantum) (Ezratty, 2025).



classical technologies

QRA

quantum resistant cryptography

classical cryptography resisting to quantum algorithms

PQC

post-quantum cryptography

new classical asymmetric keys and signatures resisting to quantum algorithms

symmetric keys

classical cryptography already resistant to quantum algorithms (AES, ...)

unsafe cryptography

classical cryptography based on RSA

Figure 4: Various types of quantum communication and cybersecurity technologies (classical) (Ezratty, 2025).

Quantum key distribution: BB84

1. Alice creates a random bit (0 or 1) and randomly selects one of her two basis sets, ($Z = \{|0\rangle, |1\rangle\}$ or $X = \{|+\rangle, |-\rangle\}$) to transmit her information to Bob using the quantum channel.
2. This process is repeated with Alice recording the state, basis and time of each photon sent.
3. As Bob does not know the basis the photons were encoded in, he randomly selects a basis (Z or X) to measure. He does this for each photon he receives, recording the time, measurement basis used and result.
4. After Bob has measured the photons, Alice broadcasts the basis each photon was in, and Bob broadcasts the basis each photon was being measured.
5. They discard the photons where Bob used a different basis (half on average).
6. If more than p bits differ, they abort the key and try again with a different quantum channel.

Alice's random bit	0	1	1	0	1	0	0	1
Alice's random qubit	$\|0\rangle$	$\|1\rangle$	$\|-\rangle$	$\|0\rangle$	$\|-\rangle$	$\|+\rangle$	$\|+\rangle$	$\|1\rangle$
Bob's random measuring basis	Z	X	X	X	Z	X	Z	Z
Bob's result	$\|0\rangle$	$\|+\rangle$	$\|-\rangle$	$\|+\rangle$	$\|1\rangle$	$\|+\rangle$	$\|1\rangle$	$\|1\rangle$
Shared secret key	0		1			0		1

Table 1: Example of BB84.

Quantum sensing

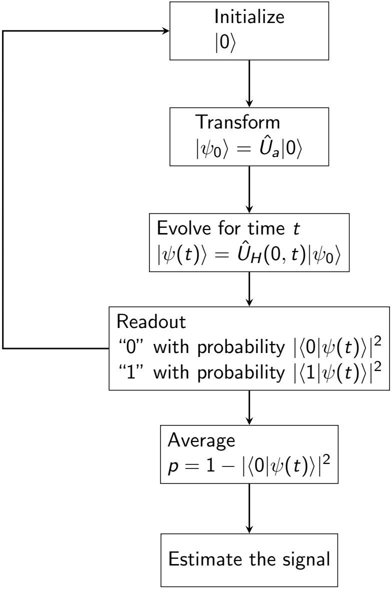
Quantum sensing[1] is typically used to describe one of the followings:

1. Use of a quantum object to measure a physical quantity (classical or quantum). The quantum object is characterized by quantized energy levels.
2. Use of quantum coherence (wave-like spatial or temporal superposition states) to measure a physical quantity.
3. Use of quantum entanglement to improve the sensitivity or precision of a measurement, beyond what is possible classically.

[1] Degen, Reinhard & Cappellaro. Quantum sensing. Rev. Mod. Phys. 89, 035002, 2017.

In analogy to DiVincenzo criteria for quantum computing, a set of attributes for quantum sensing can be defined:

1. The quantum system has discrete, resolvable energy levels.
2. It must be possible to initialize the quantum system into a well-known state and to read out its state.
3. The quantum system can be coherently manipulated, typically by time-dependent fields.
4. The quantum system interacts with a relevant physical quantity, quantified by a coupling parameter, and will lead to a shift of the quantum system's energy levels or to transition between energy levels.



Initialize
$|0\rangle$

Transform
$|\psi_0\rangle = \hat{U}_a|0\rangle$

Evolve for time t
$|\psi(t)\rangle = \hat{U}_H(0,t)|\psi_0\rangle$

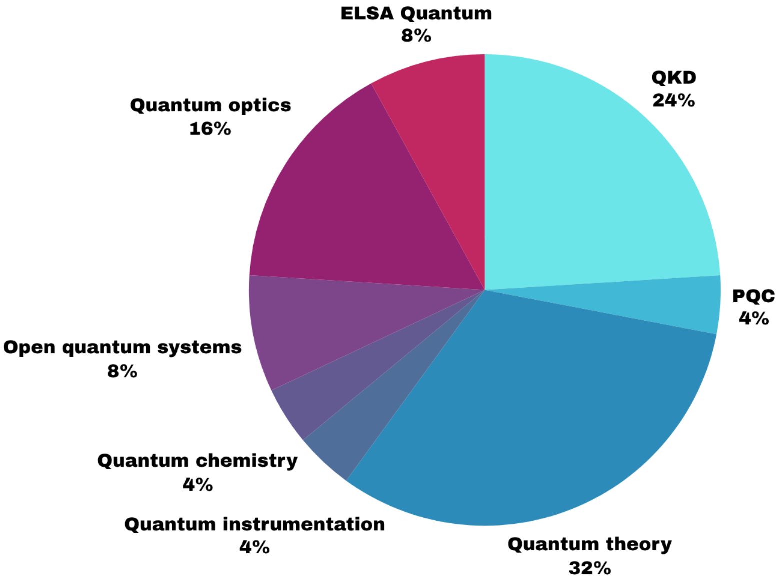
Readout
"0" with probability $|\langle 0|\psi(t)\rangle|^2$
"1" with probability $|\langle 1|\psi(t)\rangle|^2$

Average
$p = 1 - |\langle 0|\psi(t)\rangle|^2$

Estimate the signal

Figure 5: Basic steps of the quantum sensing process.

Malaysia’s quantum research landscape



Classical bits

Imagine an unfair coin. The probability of getting a head is $P(C = H) = p$, the probability of getting a tail is $P(C = T) = 1 - p$.

It can be represented as a probability table:

	$P(C)$
H	p
T	$1 - p$

More compactly, one can represent it as a column matrix,

$$\begin{aligned} P(C) &= \begin{pmatrix} p \\ 1 - p \end{pmatrix} \\ &= p \begin{pmatrix} 1 \\ 0 \end{pmatrix} + (1 - p) \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{aligned} \tag{1}$$

If we want to transform into different physical systems with different probability vectors, we can apply stochastic matrices on the probability vectors. A stochastic matrix S satisfies the following conditions to preserve the properties of probability vectors:

1. Every matrix elements are non-negative;
2. The sum of every matrix elements in a column is equals to 1.

The matrix element S_{ij} represents the probability of moving from i to j, $P(j|i)$.

Example of a 2×2 stochastic matrix $S = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}$, where $0 \leq p \leq 1$.

Consider two independent events, for example two coin tosses C_1 and C_2. The probability table of C_1 and C_2 can be given as follow.

	$P(C_1)$
H	p
T	$1-p$

	$P(C_2)$
H	q
T	$1-q$

The combined probability table of two coin tosses can be given as follows.

	$P(C_1C_2)$
HH	pq
HT	$p(1-q)$
TH	$(1-p)q$
TT	$(1-p)(1-q)$

Or, written as a probability vector,

$$P(C_1C_2) = \begin{pmatrix} pq \\ p(1-q) \\ (1-p)q \\ (1-p)(1-q) \end{pmatrix}. \tag{2}$$

Since C_1 and C_2 are independent events, $P(C_1|C_2) = P(C_1)$ and $P(C_2|C_1) = P(C_2)$, i.e. the outcome of event C_1 (C_2) is independent of event C_2 (C_1).

Also, note that $P(HH) \times P(TT) = P(HT) \times P(TH)$.

In other words, if C_1 and C_2 are not independent events, then $P(HH) \times P(TT) \neq P(HT) \times P(TH)$. C_1 and C_2 are correlated in this scenario.

One example of dependent events is drawing two cards from a deck without replacement.

Complex numbers

We denote the symbol $i = \sqrt{-1}$ with the understanding that $i^2 = -1$ to represent imaginary unit.

Any constant c multiplying with the imaginary unit is called imaginary number. For example, $12i$.

The combination of real and imaginary number is called complex number.

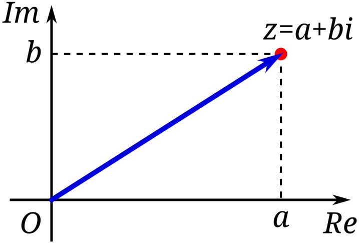


Figure 6: Argand diagram

The rectangular form, $z = a + bi$, can be rewritten into the polar form,

$$z = a + bi = r(\cos\theta + i\sin\theta) = re^{i\theta}. \tag{3}$$

Note that $Re(z) = a$ and $Im(z) = b$.

The complex conjugate of z is defined as $\bar{z} = a - bi = re^{-i\theta}$.

The modulus or absolute value of z is defined as $|z| = r = \sqrt{a^2 + b^2}$.

Hence, $|z| = \sqrt{z\bar{z}}$.

Linear algebra

A vector can be seen as a geometric entity (arrow in a coordinate system) or a set of numbers, with components relative to a coordinate system. Mathematically, a vector can be represented as a column matrix. For a two-dimensional vector $\vec{v}$,

$$\begin{aligned} \vec{v} &= \begin{pmatrix} x \\ y \end{pmatrix} && (4) \\ &= x \begin{pmatrix} 1 \\ 0 \end{pmatrix} + y \begin{pmatrix} 0 \\ 1 \end{pmatrix} && (5) \\ &= x\vec{e}_x + y\vec{e}_y && (6) \end{aligned}$$

We call $\vec{e}_x$ and $\vec{e}_y$ as the unit vectors along x and y directions respectively.

For complex vector spaces, x and y are complex numbers.

The conjugate transpose operation of a vector $\vec{v}$ is denoted by the dagger symbol † and defined by

$$\vec{v}^\dagger = \begin{pmatrix} \bar{x} & \bar{y} \end{pmatrix} \tag{7}$$

The inner product is defined as the multiplication between $\vec{v}^\dagger$ and $\vec{v}$, i.e.

$$\begin{aligned} \vec{v}^\dagger \vec{v} &= \begin{pmatrix} \bar{x} & \bar{y} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\ &= |x|^2 + |y|^2 \end{aligned} \tag{8}$$

The outer product (or tensor product) is defined as the multiplication between $\vec{v}$ and $\vec{v}^\dagger$, i.e.

$$\begin{aligned} \vec{v} \otimes \vec{v}^\dagger &= \begin{pmatrix} x \\ y \end{pmatrix} \otimes \begin{pmatrix} \bar{x} & \bar{y} \end{pmatrix} \\ &= \begin{pmatrix} |x|^2 & x\bar{y} \\ \bar{x}y & |y|^2 \end{pmatrix} \end{aligned} \tag{9}$$

More generally, tensor product is done with Kronecker product operation. For example,

$$\begin{aligned}\vec{v} \otimes \vec{v}^{\dagger} &= \begin{pmatrix} x \\ y \end{pmatrix} \otimes \begin{pmatrix} \bar{x} & \bar{y} \end{pmatrix} \\ &= \begin{pmatrix} x \begin{pmatrix} \bar{x} & \bar{y} \end{pmatrix} \\ y \begin{pmatrix} \bar{x} & \bar{y} \end{pmatrix} \end{pmatrix} \\ &= \begin{pmatrix} |x|^2 & x\bar{y} \\ \bar{x}y & |y|^2 \end{pmatrix}\end{aligned}$$

Kronecker product is not the same as matrix multiplication. For example,

$$\begin{aligned}\begin{pmatrix} a & b \\ c & d \end{pmatrix} \otimes \begin{pmatrix} e & f \\ g & h \end{pmatrix} &= \begin{pmatrix} a \begin{pmatrix} e & f \\ g & h \end{pmatrix} & b \begin{pmatrix} e & f \\ g & h \end{pmatrix} \\ c \begin{pmatrix} e & f \\ g & h \end{pmatrix} & d \begin{pmatrix} e & f \\ g & h \end{pmatrix} \end{pmatrix} \\ &= \begin{pmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{pmatrix}\end{aligned}$$

Consider a matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ with complex entries.

The transpose of a matrix A is given as

$$A^T = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^T = \begin{pmatrix} a & c \\ b & d \end{pmatrix}. \tag{10}$$

A Hermitian matrix is defined as $A = (\bar{A})^T = A^\dagger$.

The inverse matrix of A is written as A^{-1}.

The matrix multiplication between a matrix with its inverse, $AA^{-1} = A^{-1}A = I$, where I is the identity matrix.

A unitary matrix is defined as $A^\dagger = A^{-1}$.

Quantum bits

We use a different notation for vectors. Let $|\psi\rangle$ be a vector in complex vector space $\mathbb{C}^2$,

$$
\begin{aligned}
|\psi\rangle &= \psi_0|0\rangle + \psi_1|1\rangle && (11)\\
&= \begin{pmatrix}\psi_0\\ \psi_1\end{pmatrix} && (12)
\end{aligned}
$$

, where $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$ and $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$, ψ_0 and ψ_1 are complex numbers called probability amplitudes.

$|\psi\rangle$ is called a ket vector. The dual is a bra vector,

$$
\begin{aligned}
\langle\psi| &= \bar{\psi}_0\langle 0| + \bar{\psi}_1\langle 1| && (13)\\
&= \begin{pmatrix}\bar{\psi}_0 & \bar{\psi}_1\end{pmatrix} && (14)
\end{aligned}
$$

, where $\langle 0| = \begin{pmatrix}1 & 0\end{pmatrix}$ and $\langle 1| = \begin{pmatrix}0 & 1\end{pmatrix}$.

The probability of getting $|0\rangle$ is $|\psi_0|^2$, while the probability of getting $|1\rangle$ is $|\psi_1|^2$. Therefore,

$$|\psi_0|^2 + |\psi_1|^2 = \langle\psi|\psi\rangle = 1. \tag{15}$$

There are three orthonormal basis sets:

1. Z-basis: $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$
2. X-basis: $|+\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}, |-\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$
3. Y-basis: $|+i\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix}, |-i\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix}$

We use tensor product to describe composite quantum systems. For two qubits A and B, $|\psi\rangle$ and $|\phi\rangle$, the quantum state becomes

$$\begin{aligned}
|\Psi_{AB}\rangle &= |\psi\rangle \otimes |\phi\rangle \\
&= (\psi_0|0_A\rangle + \psi_1|1_A\rangle) \otimes (\phi_0|0_B\rangle + \phi_1|1_B\rangle) \\
&= \psi_0\phi_0|0_A\rangle \otimes |0_B\rangle + \psi_0\phi_1|0_A\rangle \otimes |1_B\rangle + \psi_1\phi_0|1_A\rangle \otimes |0_B\rangle \\
&\quad + \psi_1\phi_1|1_A\rangle \otimes |1_B\rangle \\
&= \psi_0\phi_0|00\rangle + \psi_0\phi_1|01\rangle + \psi_1\phi_0|10\rangle + \psi_1\phi_1|11\rangle \qquad (16) \\
&= \begin{pmatrix} \psi_0\phi_0 \\ \psi_0\phi_1 \\ \psi_1\phi_0 \\ \psi_1\phi_1 \end{pmatrix} \qquad (17)
\end{aligned}$$

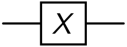
In general, a two-qubit state can be written as

$$|\psi\rangle = \psi_{00}|00\rangle + \psi_{01}|01\rangle + \psi_{10}|10\rangle + \psi_{11}|11\rangle. \qquad (18)$$

Similarly, if $\psi_{00}\psi_{11} = \psi_{01}\psi_{10}$, the two-qubit state is separable. Otherwise, the two-qubit state is entangled.

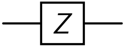
Quantum gates

Quantum circuit reads from left to right. There are several common quantum gates:-

Pauli $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, equivalently the NOT (bit-flip) gate 

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} |0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \tag{19}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} |1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \tag{20}$$

Pauli $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, equivalently the phase-flip gate 

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} |0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \tag{21}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} |1\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -\begin{pmatrix} 0 \\ 1 \end{pmatrix} = -|1\rangle \tag{22}$$

Pauli $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, a combination of Pauli X and Z gates $-\boxed{Y}-$

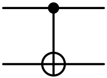
$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} |0\rangle = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = i \begin{pmatrix} 0 \\ 1 \end{pmatrix} = i|1\rangle \tag{23}$$

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} |1\rangle = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -i \begin{pmatrix} 1 \\ 0 \end{pmatrix} = -i|0\rangle \tag{24}$$

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

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} |0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = |+\rangle \tag{25}$$

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} |1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = |-\rangle \tag{26}$$

CNOT gate 

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{27}$$

$$CNOT|00\rangle = |00\rangle \tag{28}$$

$$CNOT|01\rangle = |01\rangle \tag{29}$$

$$CNOT|10\rangle = |11\rangle \tag{30}$$

$$CNOT|11\rangle = |10\rangle \tag{31}$$

Deutsch-Jozsa algorithm
1.30pm - 4.30pm GMT+8, 30 October 2025

Constant and balanced functions

Let f be a function that maps the set $\{0,1\}$ into the set $\{0,1\}$, $f:\{0,1\}\rightarrow\{0,1\}$. There are two possibilities.

A constant function gives the same output regardless of the input, i.e. $f(0)=f(1)$.

A balanced function gives an equal number of 0 and 1 as output, i.e. $f(0)\neq f(1)$.

x	$f_0(x)$	$f_1(x)$
0	0	1
1	0	1

Table 2: Constant function

x	$f_2(x)$	$f_3(x)$
0	1	0
1	0	1

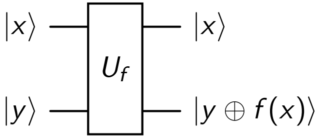
Table 3: Balanced function

Quantum oracle

A quantum oracle is a black-box that evaluates a function f. it is often represented as a unitary transformation U_f that acts on a bipartite system,

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle, \tag{32}$$

where $\oplus$ denotes addition modulo 2.



$|x\rangle$ is called the input state, $|y\rangle$ is called the ancillary state.
Show that

$$U_f^2|x\rangle|y\rangle = |x\rangle|y\rangle. \tag{33}$$

Some preliminary results

State preparation

$$\begin{aligned} H \otimes H|00\rangle &= |++\rangle \\ &= \left(\frac{1}{\sqrt{2}}\left[|0\rangle + |1\rangle\right]\right) \otimes \left(\frac{1}{\sqrt{2}}\left[|0\rangle + |1\rangle\right]\right) \\ &= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \\ &= \frac{1}{\sqrt{2^2}} \sum_{x \in \{0,1\}^2} |x\rangle \end{aligned}$$

Here, $\{0,1\}^2 = \{00, 01, 10, 11\}$.
In general,

$$H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle. \tag{34}$$

Modulo 2 arithmetic

Modulo 2 addition, $\oplus$, is also known as the XOR operation, with the following truth table:

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

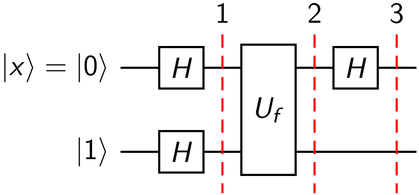
Table 4: Truth table of XOR

Let c be either 0 or 1. Find $0 \oplus c$.

Let $c_0 = 0$, $c_1 = 1$. Find $1 \oplus c_0$ and $1 \oplus c_1$. Will the result change if $c_0 = 1$, $c_1 = 0$?

Deutsch algorithm

Consider the following circuit.



Note that $|1\rangle = X|0\rangle$. For simplification, we initiate the two-qubit state as $|0\rangle|1\rangle$. At Step 1,

$$H \otimes H|0\rangle|1\rangle = |+\rangle|-\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right). \tag{35}$$

At Step 2,

$$U_f\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)=\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\oplus f(x)\rangle-|1\oplus f(x)\rangle}{\sqrt{2}}\right)$$
$$=\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|f(x)\rangle-|1\oplus f(x)\rangle}{\sqrt{2}}\right) \quad (36)$$

Regardless of the value of x, if $f(x)=0$, then

$$\frac{|f(x)\rangle-|1\oplus f(x)\rangle}{\sqrt{2}}=\frac{|0\rangle-|1\rangle}{\sqrt{2}}.$$

If $f(x)=1$, then

$$\frac{|f(x)\rangle-|1\oplus f(x)\rangle}{\sqrt{2}}=\frac{|1\rangle-|0\rangle}{\sqrt{2}}=-\frac{|0\rangle-|1\rangle}{\sqrt{2}}.$$

Combining both cases, we have

$$\frac{|f(x)\rangle-|1\oplus f(x)\rangle}{\sqrt{2}}=(-1)^{f(x)}\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right).$$

Hence, we can rewrite Equation (36) as

$$\begin{aligned}
&\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}}\right) \\
&= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left[(-1)^{f(x)}\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\right] \\
&= \frac{1}{2}\Big((-1)^{f(0)}|0\rangle|0\rangle - (-1)^{f(0)}|0\rangle|1\rangle + (-1)^{f(1)}|1\rangle|0\rangle - (-1)^{f(1)}|1\rangle|1\rangle\Big) \\
&= \left(\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \qquad (37)
\end{aligned}$$

At Step 3, we apply a Hadamard gate on the first qubit, $|x\rangle$, from Equation (37),

$$\begin{aligned}
&H \otimes I \left(\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\
&= \left(\frac{(-1)^{f(0)}(|0\rangle + |1\rangle) + (-1)^{f(1)}(|0\rangle - |1\rangle)}{2} \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\
&= \left(\frac{[(-1)^{f(0)} + (-1)^{f(1)}]|0\rangle + [(-1)^{f(0)} - (-1)^{f(1)}]|1\rangle}{2} \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)
\end{aligned} \tag{38}$$

If f is a constant function, i.e. $f(0) = f(1)$, Equation (38) becomes

$$\pm|0\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right). \tag{39}$$

If f is a balanced function, i.e. $f(0) \neq f(1)$, Equation (38) becomes

$$\pm|1\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right). \tag{40}$$

Before we go into Deutsch-Jozsa algorithm, it is useful to know that

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{x\cdot y}|y\rangle, \tag{41}$$

where $x \cdot y = x_1 y_1 \oplus x_2 y_2 \oplus \ldots \oplus x_n y_n$.
For one qubit,

$$H|0\rangle = \frac{1}{\sqrt{2}} \sum_{y=0}^{1} (-1)^{0\cdot y}|y\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle\right),$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \sum_{y=0}^{1} (-1)^{1\cdot y}|y\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle\right).$$

Or, in general,

$$H|x\rangle = \frac{1}{\sqrt{2}} \sum_{y=0}^{1} (-1)^{x\oplus y}|y\rangle.$$

Equation (41) can be shown to take its form by combining the action of two Hadamard gates on two qubits $|x_1\rangle$, $|x_2\rangle$ and generalize to n qubits,

$$
\begin{aligned}
H^{\otimes 2}|x_1\rangle|x_2\rangle &= \frac{1}{2}\left(\sum_{y_1=0}^{1}(-1)^{x_1\oplus y_1}|y_1\rangle\right)\left(\sum_{y_2=0}^{1}(-1)^{x_2\oplus y_2}|y_2\rangle\right) \\
&= \frac{1}{2}\left(\sum_{y_1=0}^{1}\sum_{y_2=0}^{1}(-1)^{x_1\oplus y_1}(-1)^{x_2\oplus y_2}|y_1\rangle|y_2\rangle\right) \\
&= \frac{1}{2}\left(\sum_{y_1=0}^{1}\sum_{y_2=0}^{1}(-1)^{x_1\oplus y_1+x_2\oplus y_2}|y_1\rangle|y_2\rangle\right)
\end{aligned}
$$

Example: Find $H^{\otimes 2}|10\rangle$.

Using Equation (41), we can rewrite Equation (38) as follow:

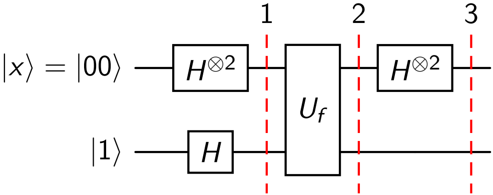
$$\begin{aligned} & H \otimes I \left(\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\ &= \frac{1}{\sqrt{2}} H \otimes I \left(\sum_{x=0}^{1} (-1)^{f(x)} |x\rangle \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\ &= \frac{1}{2} \left(\sum_{x=0}^{1} (-1)^{f(x)} \sum_{y=0}^{1} (-1)^{x \cdot y} |y\rangle \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\ &= \frac{1}{2} \left(\sum_{x=0}^{1} \sum_{y=0}^{1} (-1)^{f(x) + x \cdot y} |y\rangle \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \qquad (42) \end{aligned}$$

Verify that Equation (42) can be expanded into Equation (38), provided as follow:

$$\left(\frac{[(-1)^{f(0)} + (-1)^{f(1)}]|0\rangle + [(-1)^{f(0)} - (-1)^{f(1)}]|1\rangle}{2} \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

Deutsch-Jozsa algorithm

Consider the following example circuit.



The three-qubit state is initiated as $|00\rangle|1\rangle$. At Step 1,

$$H^{\otimes 2} \otimes H|00\rangle|1\rangle = \left(\frac{1}{2}\sum_{x\in\{0,1\}^2}|x\rangle\right) \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right). \qquad (43)$$

At Step 2,

$$\begin{aligned} & U_f\left(\frac{1}{2}\sum_{x\in\{0,1\}^2}|x\rangle\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \\ = & \left(\frac{1}{2}\sum_{x\in\{0,1\}^2}(-1)^{f(x)}|x\rangle\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \end{aligned} \tag{44}$$

We know from Deutsch algorithm that the ancillary qubit is not important. Hence, we can focus only on the input state during Step 3.

At Step 3,

$$
\begin{aligned}
&H \otimes H \left(\frac{1}{2} \sum_{x \in \{0,1\}^2} (-1)^{f(x)} |x\rangle \right) \\
&= \frac{1}{2} H \otimes H \left((-1)^{f(00)}|00\rangle + (-1)^{f(01)}|01\rangle + (-1)^{f(10)}|10\rangle + (-1)^{f(11)}|11\rangle \right) \\
&= \frac{1}{2} \Big((-1)^{f(00)}|++\rangle + (-1)^{f(01)}|+-\rangle + (-1)^{f(10)}|-+\rangle \\
&\qquad + (-1)^{f(11)}|--\rangle \Big) \\
&= \left(\frac{1}{2}\right) \left(\frac{1}{2}\right) \Big((-1)^{f(00)}[|00\rangle + |01\rangle + |10\rangle + |11\rangle] \\
&\qquad + (-1)^{f(01)}[|00\rangle - |01\rangle + |10\rangle - |11\rangle] \\
&\qquad + (-1)^{f(10)}[|00\rangle + |01\rangle - |10\rangle - |11\rangle] \\
&\qquad + (-1)^{f(11)}[|00\rangle - |01\rangle - |10\rangle + |11\rangle] \Big) \qquad (45)
\end{aligned}
$$

Verify that the following simplification can be expanded into Equation (45).

$$
\begin{aligned}
&\frac{1}{2^2}\left(\sum_{x\in\{0,1\}^2}\sum_{y\in\{0,1\}^2}(-1)^{f(x)+x\cdot y}|y\rangle\right)\\
&=\frac{1}{4}\Big([(-1)^{f(00)}+(-1)^{f(01)}+(-1)^{f(10)}+(-1)^{f(11)}]|00\rangle\\
&\quad+[(-1)^{f(00)}-(-1)^{f(01)}+(-1)^{f(10)}-(-1)^{f(11)}]|01\rangle\\
&\quad+[(-1)^{f(00)}+(-1)^{f(01)}-(-1)^{f(10)}-(-1)^{f(11)}]|10\rangle\\
&\quad+[(-1)^{f(00)}-(-1)^{f(01)}-(-1)^{f(10)}+(-1)^{f(11)}]|11\rangle\Big)
\end{aligned}
$$

If the function is constant, due to the constructive interference for $|00\rangle$, the probability of getting $|00\rangle$ is 1.

If the function is balanced, due to the destructive interference for $|00\rangle$, the probability of getting $|00\rangle$ is 0.

Shor's algorithm
9.00am - 12.00pm GMT+8, 31 October 2025

Some preliminary results

Eigenvalues and eigenvectors

Let $|\psi\rangle$ be a vector. An eigenvector is a vector that remains unchanged under a linear transformation. For a unitary transformation U, the eigenequation is given by

$$U|\psi\rangle = e^{2\pi i\omega}|\psi\rangle, \tag{46}$$

where $e^{2\pi i\omega}$ is the eigenvalue of the unitary transformation. ω is the phase of the eigenvalue.

Binary fraction

The decimal fraction allows us to express rational numbers as a fraction whose denominator is a power of ten. For example,

$$0.15625 = 1 \times 10^{-1} + 5 \times 10^{-2} + 6 \times 10^{-3} + 2 \times 10^{-4} + 5 \times 10^{-5}.$$

Similarly, the binary fraction allows us to represent the above rational number as a fraction whose denominator is a power of two,

$$0.00101 = 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5}.$$

A binary representation is useful because we can encode it using qubits.

Factorization problem

Suppose we have a dividend a, with divisor m, quotient k and remainder b. We can write the following equation,

$$a = km + b.$$

In congruence relation terminology, we can write

$$a = b \bmod m.$$

For example,

1. $1 \bmod 15 = ?$
2. $2 \bmod 15 = ?$
3. $4 \bmod 15 = ?$
4. $8 \bmod 15 = ?$
5. $16 \bmod 15 = ?$
6. $32 \bmod 15 = ?$
7. $64 \bmod 15 = ?$

Notice that there is a repetition of the outcomes after every four numbers. The cycle (or period) length r is equal to 4 in our example.

Take note that $2^4 = 1 \bmod 15$.

Then, 15 can be divided by $2^4 - 1$, i.e. $15|2^4 - 1$. We can break down the factor $2^4 - 1$ by difference of squares,

$$15|(2^2 - 1)(2^2 + 1).$$

Hence, we identified the prime factors of 15, i.e. 3 and 5.

In general,

$$N|(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1). \tag{47}$$

One caveat of this approach is that r has to be even, since $a^{\frac{r}{2}}$ needs to be an integer.

To factor $N = pq$, a factoring algorithm follows the steps below:

1. Select any number $1 < a < N$ and find the greatest common divisor (gcd) of a and N. If $\text{gcd} \neq 1$, then the it is a nontrivial common factor of a and N, hence we found one of the factors of N, $p = \text{gcd}(a, N)$. The other factor will be $q = \frac{N}{p}$.
2. If $\text{gcd} = 1$, we find the period r of $a^r \text{mod } N$. If r is odd, we go back to step 1 and pick a different a.
3. Now, $a^r = 1 \text{ mod } N$. Subtract 1 from both sides, $a^r - 1 = 0 \text{ mod } N$. This means that $a^r - 1 = kN = kpq$.
4. Factoring the left hand side, we have $(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) = kpq$.
5. Hence, $a^{\frac{r}{2}} - 1 = cp$, $a^{\frac{r}{2}} + 1 = dq$. Since each term $a^{\frac{r}{2}} - 1$ and $a^{\frac{r}{2}} + 1$ share a non-trivial factor with $N = pq$, we have thus factored N.

Quantum Fourier Transform

Quantum Fourier Transform (qFT) can be thought of as a unitary transformation with the following unitary matrix,

$$\hat{U}_{qFT} = \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} e^{\frac{2\pi ijk}{N}} |j\rangle\langle k|, \tag{48}$$

where $N = 2^n$.

Example: Let $\omega = e^{\frac{2\pi i}{N}}$. Write down the unitary matrix $\hat{U}_{qFT}$ for $N = 2^2 = 4$.

$\hat{U}_{qFT}$ acts on a quantum state $|x\rangle = \sum_{k=0}^{N-1} x_k|k\rangle$ and maps it to $|y\rangle = \sum_{j=0}^{N-1} y_j|j\rangle$, where

$$y_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi ijk}{N}} x_k. \tag{49}$$

For a single qubit, $n = 1$. $\hat{U}_{qFT}$ becomes

$$\begin{aligned}\hat{U}_{qFT} &= \frac{1}{\sqrt{2}} \sum_{j,k=0}^{1} e^{\pi ijk}|j\rangle\langle k| \\ &= \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + e^{\pi i}|1\rangle\langle 1|) \\ &= \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|)\end{aligned}$$

Therefore,

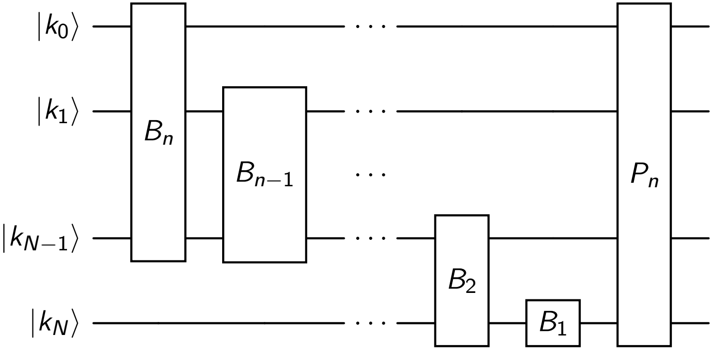
$$\begin{aligned}\hat{U}_{qFT}|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|)|0\rangle \\ &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle,\end{aligned}$$

$$\begin{aligned}\hat{U}_{qFT}|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|)|1\rangle \\ &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle,\end{aligned}$$

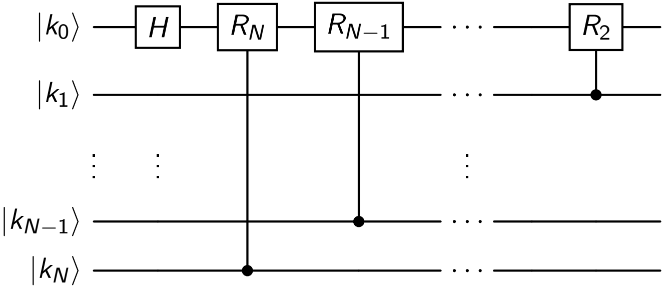
Example: Identify the general form of qFT for two qubits based on Equation (48). Hence, find the qFT of $|00\rangle$ and $|01\rangle$.

Since qFT is a unitary transformation, there exists an inverse qFT that maps $|y\rangle$ into $|x\rangle$.

In general, the qFT circuit looks like



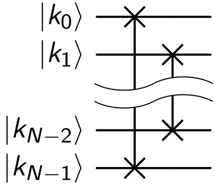
The gate $-\boxed{B_n}-$ means



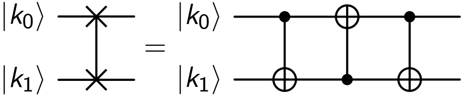
where $-\boxed{R_n}-$ is the unitary rotation,

$$R_n = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^n}} \end{pmatrix}. \tag{50}$$

The gate $-\boxed{P_n}-$ means a set of permutations of (i)-th qubit to $(N - i - 1)$-th qubit,

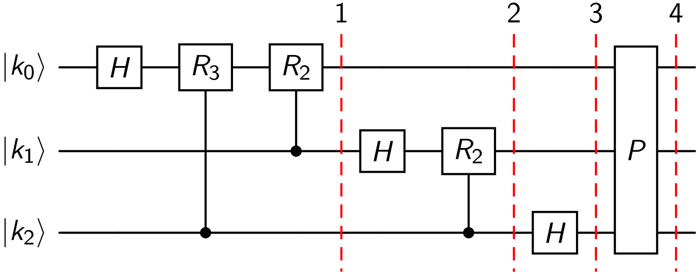


where the permutation between two qubits can be executed through 3 CNOT gates,



Note that the permutation P_n depends on how the hardware orders the qubits and sometimes it is not necessary to perform P_n.

Example: Three-qubit quantum Fourier transform



The rotation matrices are given by

$$R_2 = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^2}} \end{pmatrix}, \; R_3 = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^3}} \end{pmatrix}.$$

We note that

$$\begin{aligned} H|k_j\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{k_j}|1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + (e^{\pi i})^{k_j}|1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + (e^{\pi i k_j})|1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + (e^{2\pi i \frac{k_j}{2}})|1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + (e^{2\pi i [0.k_j]})|1\rangle\right) \qquad (51) \end{aligned}$$

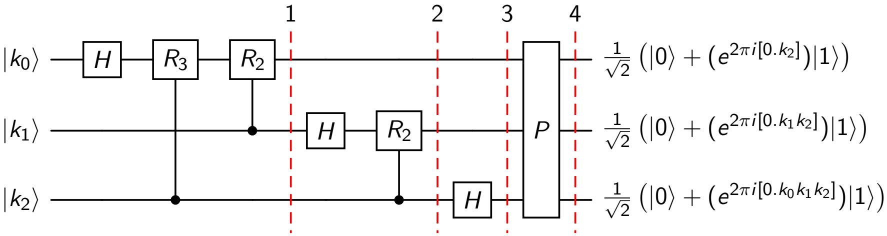
Also,

$$\begin{aligned} R_n|0\rangle &= |0\rangle, \qquad (52) \\ R_n|1\rangle &= e^{\frac{2\pi i}{2^n}}|1\rangle. \qquad (53) \end{aligned}$$

We can view each step from the example as the consequence of the B_n gates. If we understand how B_n works, we can generalize for every B_n gates. For step 1,

$$
\begin{aligned}
|k_0\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}\left(|0\rangle + (e^{2\pi i[0.k_0]})|1\rangle\right) \\
&\xrightarrow{C-R_3} \frac{1}{\sqrt{2}}\left(|0\rangle + (e^{2\pi i[0.k_0]}e^{2\pi i\frac{k_2}{2^3}})|1\rangle\right) \\
&\xrightarrow{C-R_2} \frac{1}{\sqrt{2}}\left(|0\rangle + (e^{2\pi i[0.k_0]}e^{2\pi i\frac{k_2}{2^3}}e^{2\pi i\frac{k_1}{2^2}})|1\rangle\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + (e^{2\pi i[0.k_0]}e^{2\pi i[0.00k_2]}e^{2\pi i[0.0k_1]})|1\rangle\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + (e^{2\pi i[0.k_0k_1k_2]})|1\rangle\right)
\end{aligned}
$$

Hence,


1
2
3
4
$|k_0\rangle$
$|k_1\rangle$
$|k_2\rangle$
H
R_3
R_2
H
R_2
H
P
$\frac{1}{\sqrt{2}}\left(|0\rangle + (e^{2\pi i[0.k_2]})|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + (e^{2\pi i[0.k_1k_2]})|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + (e^{2\pi i[0.k_0k_1k_2]})|1\rangle\right)$

Quantum phase estimation

The purpose of quantum phase estimation is to estimate the eigenvalue $e^{2\pi i\omega}$ of a unitary operator,

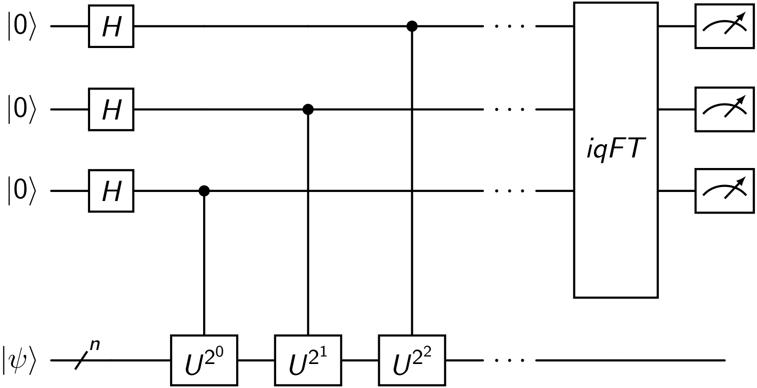
$$U|\psi\rangle = e^{2\pi i\omega}|\psi\rangle, \tag{54}$$

by preparing a quantum circuit to transform

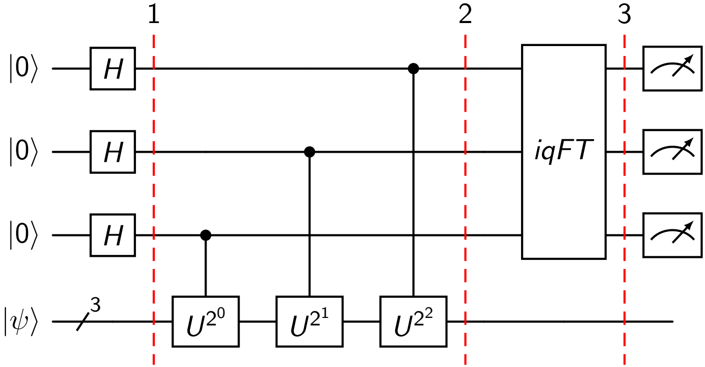
$$|\psi\rangle|0\rangle \rightarrow |\psi\rangle|\phi\rangle \tag{55}$$

and then obtaining the phase estimation by measuring $|\phi\rangle$.

A general quantum phase estimation algorithm looks like the following:



Example: Three-qubit quantum phase estimation



The unitary matrix U^{2^k} introduces the phase,

$$U^{2^k}|\psi\rangle = e^{2\pi i\omega 2^k}|\psi\rangle. \tag{56}$$

After the first step, we have

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |\psi\rangle. \tag{57}$$

At step 2, the first control-U^{2^0} will introduce a phase to $|\psi\rangle$,

$$|+\rangle^{\otimes 2} \otimes \frac{|0\rangle|\psi\rangle + e^{2\pi i[\omega]}|1\rangle|\psi\rangle}{\sqrt{2}} = |+\rangle^{\otimes 2} \otimes \frac{|0\rangle + e^{2\pi i[\omega]}|1\rangle}{\sqrt{2}} \otimes |\psi\rangle. \tag{58}$$

Following the same logic, the final state at step 2 is

$$\frac{|0\rangle + e^{2\pi i[2^2\omega]}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i[2\omega]}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i[\omega]}|1\rangle}{\sqrt{2}} \otimes |\psi\rangle \tag{59}$$

If we let $\omega = 0.k_0 k_1 k_2$, then the final state at step 2 becomes

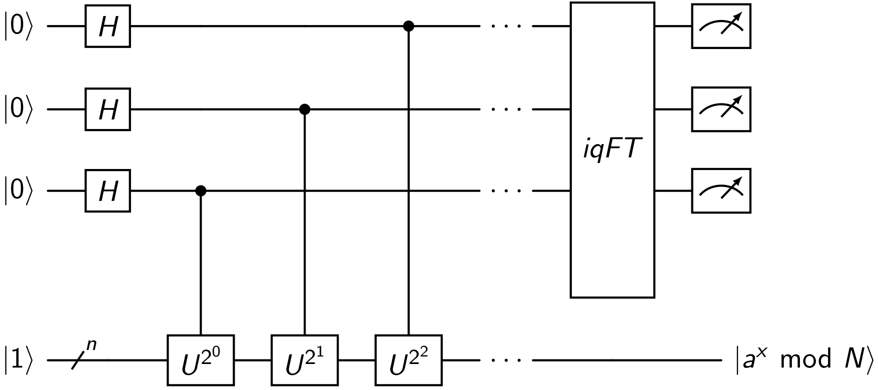
$$
\begin{aligned}
&\frac{|0\rangle + e^{2\pi i[2^2\omega]}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i[2\omega]}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i[\omega]}|1\rangle}{\sqrt{2}} \otimes |\psi\rangle \\
&= \frac{|0\rangle + e^{2\pi i[k_0 k_1 . k_2]}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i[k_0 . k_1 k_2]}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i[0.k_0 k_1 k_2]}|1\rangle}{\sqrt{2}} \otimes |\psi\rangle \\
&= \frac{|0\rangle + e^{2\pi i[0.k_2]}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i[0.k_1 k_2]}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i[0.k_0 k_1 k_2]}|1\rangle}{\sqrt{2}} \otimes |\psi\rangle
\end{aligned}
\tag{60}
$$

We note that the integers in front of the binary representation in the first equality is ignored in the second equality, because $e^{2\pi ij} = 1$ for any integer j.

This is the quantum Fourier transform that we have seen just now! By applying the inverse quantum Fourier transform, the measurement of the three qubits $|k_0 k_1 k_2\rangle$ will tell us the phase of $|\psi\rangle$.

Shor's algorithm

The complete quantum circuit of Shor's algorithm looks like the following:



The second register $|1\rangle$ uses $n = \lceil \log_2 N \rceil$ number of qubits. For the first register, $2n$ qubits will be sufficient to achieve the accuracy to find r.

In the module, we try to factor $N = 15$ by following the steps below.

1. Select any number $1 < a < 15$ and find the ones with $\gcd(a, 15) = 1$. The possible $a = \{2, 4, 7, 8, 11, 13, 14\}$.
2. Let $a = 2$. We identify the set of unitaries, $\{U^{2^0}, U^{2^1}, \ldots, U^{2^k}\}$ by modular exponentiation.
3. By phase estimation, we determine the period r of $2^r \text{mod } 15$. If r is odd, we go back to step 1 and pick a different a.
4. Now, $2^r = 1 \bmod 15$. Subtract 1 from both sides, $2^r - 1 = 0 \bmod 15$. This means that $2^r - 1 = 15k = kpq$.
5. Factoring the left hand side, we have $(2^{\frac{r}{2}} - 1)(2^{\frac{r}{2}} + 1) = kpq$.
6. Hence, $2^{\frac{r}{2}} - 1 = cp$, $2^{\frac{r}{2}} + 1 = dq$. Since each term $2^{\frac{r}{2}} - 1$ and $2^{\frac{r}{2}} + 1$ share a non-trivial factor, we have thus factored 15.

Modular exponentiation

The set $\{U^{2^0}, U^{2^1}, \ldots, U^{2^k}\}$ can be found through modular exponentiation. For $a = 2$, U^{2^k} is labeled as M_2^k in the module. For the first iteration M_2, we can study how M_2 transforms the state until it forms a complete cycle, as follows.

Original state	Binary representation	After M_2	Binary representation
$\|1\rangle$	$\|0001\rangle$	$\|2\rangle$	$\|0010\rangle$
$\|2\rangle$	$\|0010\rangle$	$\|1\rangle$	$\|0100\rangle$
$\|4\rangle$	$\|0100\rangle$	$\|8\rangle$	$\|1000\rangle$
$\|8\rangle$	$\|1000\rangle$	$\|1\rangle$	$\|0001\rangle$

This is a permutation of q_0 to q_1, q_1 to q_2, q_2 to q_3, resulting $q_0q_1q_2q_3$ to $q_1q_2q_3q_0$.

The next iteration requires us to compute the modular exponentiation M_b, where $b = 2^{2^1} \pmod{15} = 4$. Similarly, we study how M_4 transforms the state until it forms two complete cycles, as follows.

Original state	Binary representation	After M_2	Binary representation
$\lvert 1\rangle$	$\lvert 0001\rangle$	$\lvert 4\rangle$	$\lvert 0100\rangle$
$\lvert 2\rangle$	$\lvert 0010\rangle$	$\lvert 8\rangle$	$\lvert 1000\rangle$
$\lvert 4\rangle$	$\lvert 0100\rangle$	$\lvert 1\rangle$	$\lvert 0001\rangle$
$\lvert 8\rangle$	$\lvert 1000\rangle$	$\lvert 2\rangle$	$\lvert 0010\rangle$

This is a permutation of q_1 to q_3 and q_0 to q_2, resulting $q_0q_1q_2q_3$ to $q_2q_3q_0q_1$.

For the third iteration, we want to find M_b, where $b = 2^{2^2} \pmod{15} = 16 \pmod{15} = 1$. M_1 is basically an identity operator, hence we can end the iteration here. For $a = 2$, $N = 15$, only two qubits are needed.

Some possible modifications on the module to test your understanding:

1. For $N = 15$, test with $a = \{4, 7, 8, 11, 13, 14\}$.
2. Factorize $N = 21$.
3. Factorize $N = 33$.

Workshop on NISQ algorithms

Timetable

Day 1	**6 November 2025**
0800 - 0900	Registration
0900 - 1200	Introduction to variational quantum eigensolver
1200 - 1330	Lunch break
1330 - 1630	Introduction to combinatorial optimization problem
Day 2	**31 October 2025**
0900 - 1200	Introduction to quantum error correction
1200 - 1500	Lunch break and prayer time
1500 - 1545	Kwek Leong Chuan
1545 - 1630	Yanoar Pribadi Sarwono: Quantum computing for electronic structure calculations

Variational quantum eigensolver
9.00am - 12.00pm GMT+8, 6 November 2025

Qiskit: Fake Backends and AerSimulator

Simulator	Fake Backends	AerSimulator
Purpose	Mimics specific QPUs by snapshots	General purpose, high performance simulation
Noise model	Automatically applies noise model from QPU snapshots	Custom or based on real QPU calibration data
Circuit size	Limited to the capabilities of the mimicked QPU	Can handle larger circuits
Results	Moderate runtime for QPU-specific tests	Shorter runtime for a wide range of simulations
Use case	Testing transpiler and QPU-specific behavior	General development, custom noise models

Table 5: https://quantum.cloud.ibm.com/docs/en/guides/local-simulators

Fake Backends

```
from qiskit_ibm_runtime import QiskitRuntimeService
from qiskit_ibm_runtime.fake_provider import FakeManilaV2
from qiskit.transpiler.preset_passmanagers import
generate_preset_pass_manager

fake_manila = FakeManilaV2()
pm = generate_preset_pass_manager(backend=fake_manila,
optimization_level=1)
backend=fake_manila
```

AerSimulator

```
from qiskit_aer import AerSimulator

simulator = AerSimulator()
compiled_circuit = transpile(circuit, simulator)
result = simulator.run(compiled_circuit, shots=1024).result()
counts = result.get_counts()
print(counts)
```

Variational quantum eigensolver

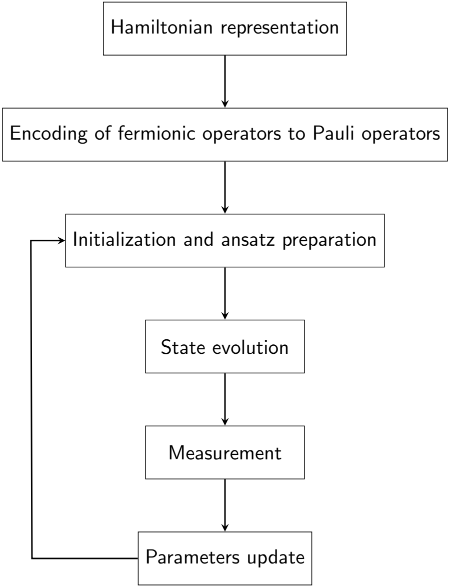
Variational quantum eigensolver (VQE) is a hybrid classical-quantum algorithm during the noisy intermediate-scale quantum (NISQ) era, in order to identify the ground state energy of a quantum system with respect to a trial wavefunction, which is called the ansatz:

$$E_{VQE} = \min_{\theta} \langle 0 | U^{\dagger}(\theta) \hat{H} U(\theta) | 0 \rangle. \tag{61}$$

The Hamiltonian $\hat{H}$ can be written as a tensor product of Pauli operators $\hat{P}$,

$$\hat{H} = \sum_{i} \omega_i \sigma_i, \tag{62}$$

where ω_i is the weight and σ_i is the Pauli matrices, $(\sigma_0, \sigma_1, \sigma_2, \sigma_3) = (I, X, Y, Z)$.



Hamiltonian representation

Encoding of fermionic operators to Pauli operators

Initialization and ansatz preparation

State evolution

Measurement

Parameters update

Figure 7: Workflow of VQE.

Hamiltonian representation

The Hamiltonian for a multielectron atom can be written as

$$\hat{H} = -\frac{\hbar^2}{2m_e}\sum_{i=1}^{Z}\nabla_i^2 - \frac{\hbar^2}{2M}\nabla_R^2 - \sum_{i=1}^{Z}\frac{Ze^2}{4\pi\epsilon_0 r_{iR}} + \sum_{i>j}\frac{e^2}{4\pi\epsilon_0 r_{ij}}. \tag{63}$$

For a molecule with N electrons and M nuclei, the Hamiltonian can be written as

$$\begin{aligned}\hat{H} = &-\frac{\hbar^2}{2m_e}\sum_{i=1}^{Z}\nabla_i^2 - \sum_{A=1}^{M}\frac{\hbar^2}{2M_A}\nabla_A^2 - \sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A e^2}{4\pi\epsilon_0 r_{iA}} + \sum_{i>j}\frac{e^2}{4\pi\epsilon_0 r_{ij}} \\ &+ \sum_{B>A}\frac{Z_A Z_B e^2}{4\pi\epsilon_0 R_{AB}}. \end{aligned} \tag{64}$$

Through Born-Oppenheimer approximation, which assumes that the wavefunction of nuclei and electrons can be treated separately, we have the electronic Hamiltonian,

$$\hat{H}_e = -\frac{\hbar^2}{2m_e}\sum_{i=1}^{N}\nabla_i^2 - \sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A e^2}{4\pi\epsilon_0 r_{iA}} + \sum_{i>j}\frac{e^2}{4\pi\epsilon_0 r_{ij}}. \tag{65}$$

Electrons are fermions that satisfy Pauli exclusion principle, i.e. two identical fermions cannot occupy the same state. The wavefunction of individual fermion can be written as the Slater determinant,

$$\Psi(r_1, r_2, \ldots, r_N) = \frac{1}{\sqrt{N!}}\begin{vmatrix} \psi_{k_1}(r_1) & \psi_{k_1}(r_2) & \ldots & \psi_{k_1}(r_N) \\ \psi_{k_2}(r_1) & \psi_{k_2}(r_2) & \ldots & \psi_{k_2}(r_N) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{k_N}(r_1) & \psi_{k_N}(r_2) & \ldots & \psi_{k_N}(r_N) \end{vmatrix}. \tag{66}$$

The second quantization (or occupation number representation) is a formalism used to describe many-body quantum systems by asking "how many particles are there in each state",

$$\Psi(r_1, r_2, \ldots, r_N) \rightarrow |k_1, k_2, \ldots, k_N\rangle, \tag{67}$$

where

$$n_i = \begin{cases} 1 \text{ if } \psi_{k_i} \text{ is occupied} \\ 0 \text{ if } \psi_{k_i} \text{ is empty} \end{cases} \tag{68}$$

Vacuum state describes a state without particles,

$$|0\rangle = |0_{k_1}, 0_{k_2}, \ldots, 0_{k_N}\rangle. \tag{69}$$

The creation ($\hat{a}^\dagger$) and annihilation ($\hat{a}$) operators are being introduced in the second quantization,

$$\begin{aligned} \hat{a}|0\rangle &= 0, && (70)\\ \hat{a}|1\rangle &= |0\rangle, && (71)\\ \hat{a}^\dagger|0\rangle &= |1\rangle, && (72)\\ \hat{a}^\dagger|1\rangle &= 0. && (73) \end{aligned}$$

The order in which the operators are applied is important,

$$|k_1,\, k_2,\, \ldots,\, k_N\rangle = (\hat{a}^\dagger)^{k_1}(\hat{a}^\dagger)^{k_2}\ldots(\hat{a}^\dagger)^{k_N}|0\rangle.$$

The Hamiltonian from Equation (65) can then be rewritten into single- and two-body operators by using the creation and annihilation operators,

$$\hat{H} = \sum_{ij} h_{ij} \hat{a}_i^\dagger \hat{a}_j + \frac{1}{2} \sum_{ijkl} h_{ijkl} \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_k \hat{a}_l, \tag{74}$$

where

$$h_{ij} = \int \bar{\phi}_i(x) \left[-\frac{\nabla^2}{2} - \sum_{\alpha} \frac{Z_\alpha}{r_{\alpha,x}} \right] \phi_j(x) dx, \tag{75}$$

$$h_{ijkl} = \int \int \bar{\phi}_i(x_1) \bar{\phi}_j(x_2) \left[\frac{1}{r_{ij}} \right] \phi_k(x_1) \phi_l(x_2) dx_1 dx_2. \tag{76}$$

Encoding of fermionic operators to Pauli operators

There are a variety of mapping schemes to transform fermionic operators into spin operators (qubits). One of the conventional approach is through Jordan-Wigner transformation,

$$\hat{a}_j^\dagger = \frac{1}{2}(X_j - iY_j) \otimes_{k<j} Z_k, \tag{77}$$

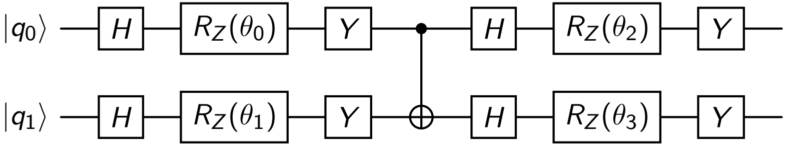
$$\hat{a}_j = \frac{1}{2}(X_j + iY_j) \otimes_{k<j} Z_k. \tag{78}$$

The string of Z gates introduces the required phase change of -1 if the parity of set of qubits with index less than i is 1 (odd), and do nothing if the parity is 0 (even).

This ensures the qubit operations follow the anti-commutation relations of $\hat{a}_j^\dagger$ and $\hat{a}_j$.

Ansatz preparation

Ansatz is a parametrized quantum circuit for which the parameters can be updated after each run.



The right choice of ansatz is critical to obtain a final solution close to the true state of interest.

It is essential to maximize the span of an ansatz in parts of the Hilbert space that contain the solution, which is the expressibility of the ansatz.

The trainability of an ansatz refers to the ability to find the best set of parameters of the ansatz by repeated optimization in a tractable time.

Barren plateau

An ansatz is trainable if its expected gradient vanishes at most polynomially as a function of the various metrics of the problem. If the gradient vanishes exponentially, the ansatz suffers from the barren plateau problem.

Barren plateau is akin to the vanishing gradient problem in machine learning. For NISQ algorithm applications, the situation is different from two aspects:

1. The estimation of the gradients on a quantum device is essentially stochastic;
2. Barren plateau problem depends on the system size, expressibility of the ansatz, degree of entanglement, quantum noise, or randomized initialization of ansatz.

Quantum approximate optimization algorithm

1.30pm - 4.30pm GMT+8, 6 November 2025

Quadratic unconstrained binary optimization problem

Quadratic unconstrained binary optimization (QUBO) problem is an NP-hard combinatorial optimization problem.

QUBO problem tries to minimize a binary cost function $f(x)$,

$$\min_x f(x) = \min_x x^T Q x = \min_x x^T Q' x + q^T x, \tag{79}$$

where $x \in \{0, 1\}^n$ is a column vector, $Q,\ Q' \in \mathbb{R}^{n\times n}$ is a symmetric or upper triangular matrix, and $q \in \mathbb{R}^n$ is a column vector, and $Q = Q' + \mathrm{diag}(q)$.

The minimization problem can be turned into maximization by writing $\max_x - f(x)$.

Example: Minimize $y = -5x_1 - 3x_2 - 8x_3 - 6x_4 + 4x_1x_2 + 8x_1x_3 + 2x_2x_3 + 10x_3x_4$, where $x_i \in \{0, 1\}$ is binary.

This example contains a linear term $-5x_1 - 3x_2 - 8x_3 - 6x_4$ and a quadratic term $4x_1x_2 + 8x_1x_3 + 2x_2x_3 + 10x_3x_4$.

Since x_i is binary, $x_i = x_i^2$. Hence, $-5x_1 - 3x_2 - 8x_3 - 6x_4 = -5x_1^2 - 3x_2^2 - 8x_3^2 - 6x_4^2$.

Therefore, one can rewrite the QUBO problem into the following form,

$$\text{Minimize } y = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix} \begin{pmatrix} -5 & 2 & 4 & 0 \\ 2 & -3 & 1 & 0 \\ 4 & 1 & -8 & 5 \\ 0 & 0 & 5 & -6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \tag{80}$$

$$\Rightarrow \text{Minimize } y = x^T Q x \tag{81}$$

Max Cut problem

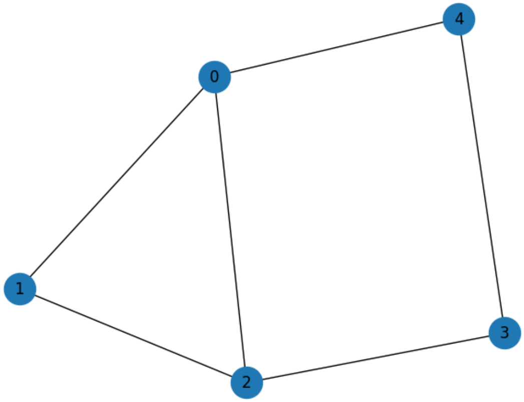
Given an undirected graph $G(V, E)$, with V as the vertex set, E as the edge set, the Max Cut problem tries to partition the vertex set into two sets such that the number of edges between the two sets is as large as possible.

Let $x_i = 1$ if vertex i is in the set, $x_i = 0$ if vertex i is not in the set (i.e. it is in the other set). The cut severs the edge that joins these two sets. Mathematically, one can test if the edge between vertex i and j is in the cut by

$$x_i + x_j - 2x_i x_j = \begin{cases} 1 & \text{if edge } (i, j) \text{ is in the cut} \\ 0 & \text{if edge } (i, j) \text{ is not in the cut} \end{cases} \tag{82}$$

Based on the graph below, identify x and Q of the following equation.

$$y = x^T Q x \tag{83}$$

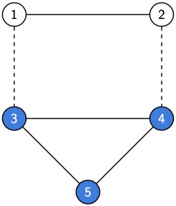


Answer:

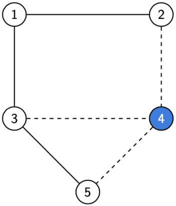
$$\begin{aligned}
&x_0 + x_1 - 2x_0x_1 + \\
&x_0 + x_2 - 2x_0x_2 + \\
&x_0 + x_4 - 2x_0x_4 + \\
&x_1 + x_2 - 2x_1x_2 + \\
&x_2 + x_3 - 2x_2x_3 + \\
&x_3 + x_4 - 2x_3x_4
\end{aligned}$$

Therefore, the Max Cut problem can be formulated into a QUBO problem by

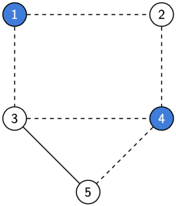
$$\text{maximize } y = \begin{pmatrix} x_0 & x_1 & x_2 & x_3 & x_4 \end{pmatrix} \begin{pmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \tag{84}$$



Cut Size: 2



Cut Size: 3



Cut Size: 5

More often than not, we are interested in constrained optimization problem. For instance, we may consider the following problem:

Minimize $y = x^T Qx$
subject to the constraint $x_1 + x_2 \leq 1$

Such a constrained optimization problem may be converted into a QUBO problem by adding a quadratic penalty term Px_1x_2, so that the QUBO formulation now becomes

$$\text{Minimize } y = x^T Qx + Px_1x_2. \tag{85}$$

Effectively, when we try to identify the optimal solution, we avoid $x_1 = x_2 = 1$, since this solution set will introduce a large positive term into the objective function y.

Quantum approximate optimization algorithm

QUBO formulation is equivalent to the Ising model. Suppose that the cost function $y = f(x)$ corresponds to the eigenvalue of a Hamiltonian $\hat{H}_C$ with an eigenstate $|x\rangle$,

$$\hat{H}_C|x\rangle = f(x)|x\rangle. \tag{86}$$

Given that $Z_i = I \otimes \ldots \otimes Z_i \otimes \ldots \otimes I$, we observe that

$$\begin{aligned} Z_i|x\rangle &= (-1)^{x_i}|x\rangle \\ &= (1 - 2x_i)|x\rangle, \end{aligned} \tag{87}$$

where $x_i \in \{0, 1\}$. In other words,

$$\frac{I - Z_i}{2}|x\rangle = x_i|x\rangle. \tag{88}$$

The cost function

$$\begin{aligned} y = f(x) &= x^T Q' x + q^T x \\ &= \sum_{ij} x_i Q'_{ij} x_j + \sum_i q_i x_i \end{aligned} \tag{89}$$

can be rewritten into the Hamiltonian of an Ising model,

$$\begin{aligned} \hat{H}_C &= \sum_{ij} Q'_{ij} \left(\frac{I - Z_i}{2} \right) \left(\frac{I - Z_j}{2} \right) + \sum_i q_i \left(\frac{I - Z_i}{2} \right) \\ &= \frac{1}{4} \sum_{ij} Q'_{ij} (I - Z_i - Z_j + Z_i Z_j) + \frac{1}{2} \sum_i q_i (I - Z_i) \\ &= \frac{1}{4} \sum_{ij} Q'_{ij} Z_i Z_j - \frac{1}{2} \sum_i \left(q_i + \sum_j Q'_{ij} \right) Z_i + \left(\frac{1}{4} \sum_{ij} Q'_{ij} + \frac{1}{2} \sum_i q_i \right) \end{aligned} \tag{90}$$

We take note that because Q' is a symmetric matrix,

$$\sum_i \left(\sum_j Q'_{ij} \right) Z_i = \sum_j \left(\sum_i Q'_{ij} \right) Z_j .$$

Also, the last term $\left(\frac{1}{4} \sum_{ij} Q'_{ij} + \frac{1}{2} \sum_i q_i \right)$ from Equation (90) is a constant, which does not affect the minimization process.

Therefore, a QUBO problem can be converted to an equivalent Ising model with Hamiltonian

$$\hat{H}_C = \frac{1}{4} \sum_{ij} Q'_{ij} Z_i Z_j - \frac{1}{2} \sum_i \left(q_i + \sum_j Q'_{ij} \right) Z_i \tag{91}$$

with a constant offset of $\left(\frac{1}{4} \sum_{ij} Q'_{ij} + \frac{1}{2} \sum_i q_i \right)$.

The corresponding unitary operator from the Hamiltonian $\hat{H}_C$ is given by the exponentiation,

$$\hat{U}(\gamma) = e^{-i\hat{H}_C\gamma}. \tag{92}$$

The exponentiation of matrix A is given by the power series

$$e^A = \sum_{i=0}^{\infty} \frac{1}{i!} A^i. \tag{93}$$

For the exponentiation of two matrices A and B, $e^{A+B} \neq e^A e^B$. We apply the Lie-Trotter product formula,

$$e^{A+B} = \lim_{n\to\infty} \left(e^{\frac{A}{n}} e^{\frac{B}{n}} \right)^n, \tag{94}$$

and implement an approximate unitary

$$\hat{U}(\gamma) \approx \prod_{ij} R_{Z_i Z_j}\left(\frac{Q_{ij}}{4}\right) \prod_{i} R_{Z_i}\left(\frac{q_i + \sum_j Q'_{ij}}{2}\right). \tag{95}$$

At this point, we successfully convert the QUBO formulation into the Hamiltonian of an Ising model. The objective is to begin with the ground state of a known system $|\psi\rangle$, evolve the system into the ground state of the Hamiltonian that we want to minimize $\hat{H}_C$ through $\hat{U}(\gamma)$, i.e.

$$\text{minimize } \langle\psi(\gamma)|\hat{H}_C|\psi(\gamma)\rangle = \text{minimize } \langle\psi|\hat{U}^\dagger(\gamma)\hat{H}_C\hat{U}(\gamma)|\psi\rangle. \tag{96}$$

However, since $\hat{U}(\gamma)$ consists of only the Pauli Z, its unitary evolution will always prepare $|\psi(\gamma)\rangle$ into the eigenstates of Pauli Z.

If we begin with the ground state of a known system $|\psi\rangle$ as the eigenstate of Pauli Z, since $\hat{U}(\gamma)$ and $\hat{H}_C$ commute with each other, $\langle\psi|\hat{U}^\dagger(\gamma)\hat{H}_C\hat{U}(\gamma)|\psi\rangle = \langle\psi|\hat{H}_C|\psi\rangle$ and we will remain in the state $|\psi\rangle$.

To counter with this effect, we introduce a mixer Hamiltonian $\hat{H}_M$,

$$\hat{H}_M = \sum_i X_i, \tag{97}$$

which will provide us with another unitary operator,

$$\hat{U}(\beta) = e^{-i\hat{H}_M\beta} \approx \prod_i^n R_x(2\beta). \tag{98}$$

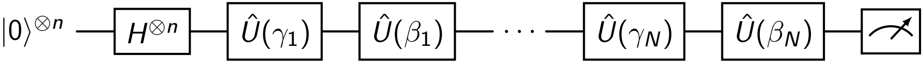
The mixer Hamiltonian can be modified into different forms, as long as it does not commute with the cost Hamiltonian $\hat{H}_C$.

The quantum approximate optimization algorithm (QAOA) consists of the following steps:

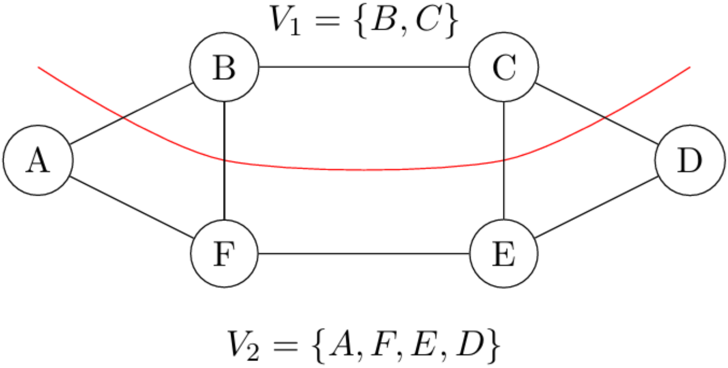
1. Define the cost Hamiltonian $\hat{H}_C$ based on the QUBO formulation.
2. Define the mixer Hamiltonian $\hat{H}_M$.
3. Define the oracles, $\hat{U}(\gamma)$ and $\hat{U}(\beta)$.
4. Prepare the initial state through state preparation.
5. Repeatedly apply the oracles $\hat{U}(\gamma)$ and $\hat{U}(\beta)$,

$$\hat{U}(\gamma,\, \beta) = \prod_{i=1}^{N} \hat{U}(\gamma_i)\hat{U}(\beta_i). \tag{99}$$

6. Optimize the parameters γ, β using classical optimizer until an approximate optimal solution is obtained.



Formulate the QUBO problem for the following Max Cut problem, and solve it by modifying the provided QAOA module. The maximum cut is 4.



Quantum error correction
9.00am - 12.00pm GMT+8, 7 November 2025

Noisy intermediate-scale quantum

Noisy intermediate-scale quantum (NISQ) is a terminology coined by Preskill to communicate with researchers and non-researchers about the current quantum computing technologies.

It informs us about the landscape that we are working on in quantum computing.

The barrier:-

- If we want to store information reliably, we need to isolate the qubits from environment;
- If we want to process the information, we need to strongly interact with the qubits.

Quantum error correction is a technique used to protect a quantum computer from errors by encoding it with highly entangled states.

Physical qubits refer to qubits that are unprotected by error correction.

Logical qubits refer to qubits which are protected by error correction.

Additionally, for NISQ algorithms, it is recommended to use shallow circuits to prevent errors from cascading within the circuit.

Fault-tolerant quantum computers refer to a framework with logical qubits by quantum error correction and proper design of quantum circuits to avoid errors cascading through the circuits.

There are a few types of errors, i.e.

1. Gate errors,
2. Decoherence error,
3. Crosstalk error.

Repetition code

Classical computers and digital data suffer from bit flip errors. The simplest error correction is to copy the bit 2 times and use majority rule to identify and fix the error. For example, we copy a single bit two times to form a majority rule, i.e. $0 \to 000$ and $1 \to 111$.

This repetition code can fix one bit flip error only. Formally, we write the three-bit repetition code as $[n, k, d] = [3, 1, 3]$, where n represents the number of bits, k is the number of encoded bits, $d = 2t + 1$ refers to the Hamming distance, where t is the maximum number of errors the code can correct.

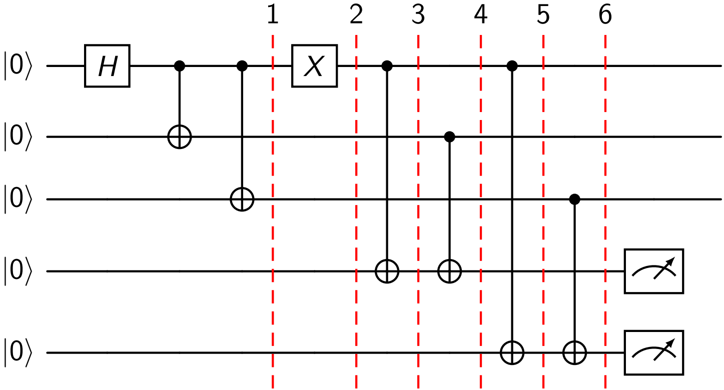
For three-bit repetition code, it is possible to detect two bit flip errors, and correct one bit flip errors.

In quantum computing, besides bit flip error X, we also have phase flip error Z.

Additionally, there are two obstacles in quantum counterpart:-

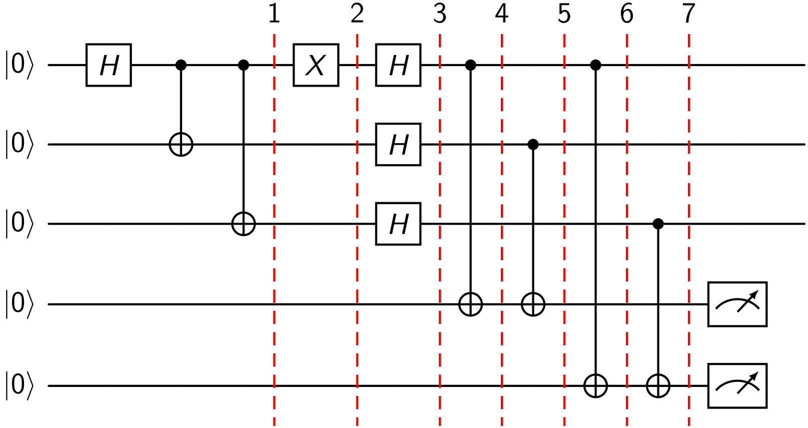
- No-cloning theorem;
- Any measurement will collapse the quantum state.

Example: Bit-flip repetition code



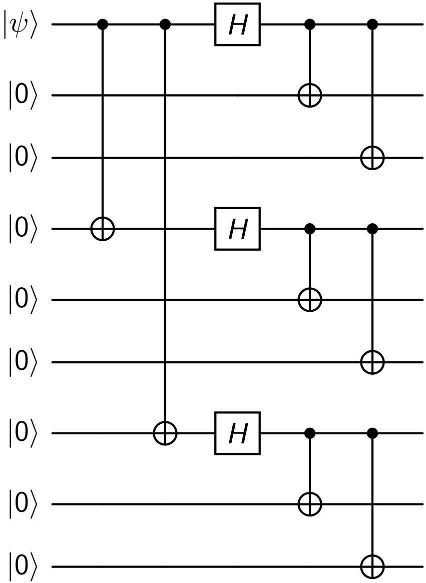
Answer:

Example: Phase-flip repetition code



Answer:

Shor's code


$|\psi\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
H
H
H

Stabilizer formalism

A stabilizer is a set of operators that leave a quantum state unchanged,

$$S|\psi\rangle = |\psi\rangle. \tag{100}$$

For example, the stabilizer for $|\psi\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ is $\{X_1X_2X_3\}$, i.e.

$$X_1X_2X_3\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle). \tag{101}$$

The code space $\mathcal{C}$ is a vector subspace of the Hilbert space, spanned by some orthogonal codewords. For three qubits, the Hilbert space is given by $\mathcal{H} \cong \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$ and the codespace is spanned by the set $\mathcal{C} =$ span $\{|000\rangle, |111\rangle\} \subset \mathcal{H}$.

We also have the error subspace, $\mathcal{F} = \mathcal{C}^{\perp} \subset \mathcal{H}$.

As the error subspace is mutually orthogonal to the codespace, we can identify which subspace the logical qubit occupies through projective measurement.

If a bit flip happened, we perform a parity measurement $Z_1 Z_2$ and $Z_2 Z_3$.

Error	$Z_1 Z_2 I$	$Z_1 I Z_3$	$I Z_2 Z_3$
III	$+1$	$+1$	$+1$
$X_1 II$	-1	-1	$+1$
$I X_2 I$	-1	$+1$	-1
$II X_3$	$+1$	-1	-1

Syndrome	State	Correction
$(+1, +1)$	$\|000\rangle$	III
$(+1, -1)$		
$(-1, +1)$		
$(-1, -1)$		

For the encoded qubits, one must also define the encoded logical operations.

e.g. For $|\vec{0}\rangle = |000\rangle$ and $|\vec{1}\rangle = |111\rangle$, the encoded operations are $\hat{X} = XXX$ and $\hat{Z} = ZII$.

Encoded logical operations must commute with elements of stabilizers.