Introduction for Quantum Algorithms for Scientific Computation: An Applied Math Perspective

Di Fang

Department of Mathematics
Duke Quantum Center
Duke University

Duke Summer School, August 2023

Outline

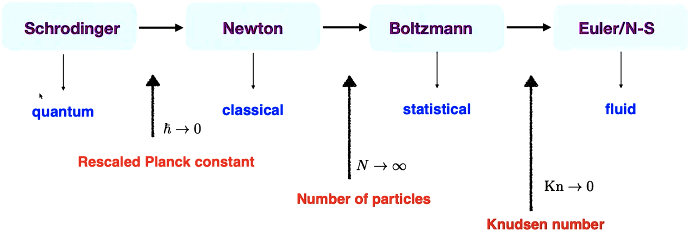
1. Intro to Quantum Computing
   - Motivations
   - Basics of QC
2. Block Encoding and Hamiltonian Simulation
3. Other advanced topics
   - General Differential Equations *(optional)*
   - Ham. Sim. with Unbounded Operators *(workshop talk)*

Part 1.1: Some **Motivations** for Quantum Computations

Different Levels of Physics

multiscale physics fig by Prof. Qin Li

Different Levels of Physics



multiscale physics fig by Prof. Qin Li

Different Levels of Physics

Schrodinger $\rightarrow$ Newton $\rightarrow$ Boltzmann $\rightarrow$ Euler/N-S

quantum classical statistical fluid

$\hbar \to 0$

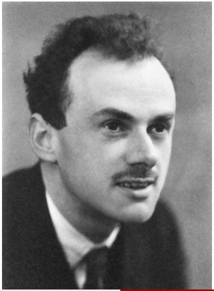
Rescaled Planck constant

$N \to \infty$

Number of particles

$\mathrm{Kn} \to 0$

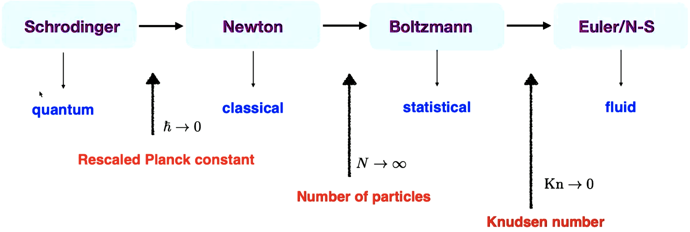
Knudsen number

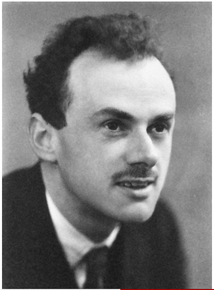


"the underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known."

Paul A. M. Dirac (1929)

Different Levels of Physics

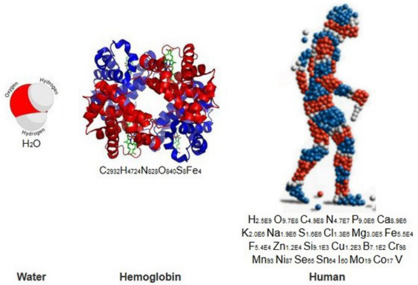




“the underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble.”

Paul A. M. Dirac (1929)

Schrödinger equation for Molecular Dynamics

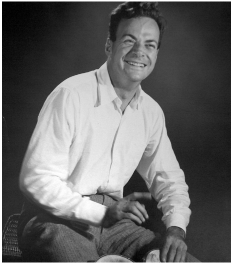


To describe its behaviour: (x: nuclei coordinates, y: electronic coordinates, M: mass of a nucleus, m: mass of an electron.)

$$\hat{H}_{\text{total}} = -\frac{\hbar^2}{2M}\Delta_x - \frac{\hbar^2}{2m}\Delta_y + V(x,y), \quad x \in \mathbb{R}^d, y \in \mathbb{R}^n$$

$$i\hbar\partial_t\psi = \hat{H}_{\text{total}}\psi$$

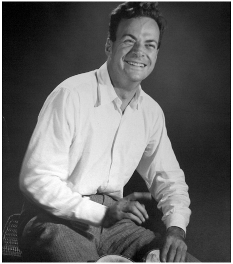
Quantum Computing 101



"... nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

Richard Feynman (1981)

Quantum Computing 101

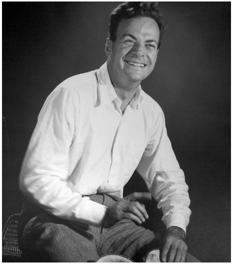


"... nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

Richard Feynman (1981)

Hamiltonian Simulation Problem (original motivation for quantum computers): Given a description of the Hamiltonian $H(t)$, an evolution time t and an initial state $|\psi(0)\rangle$, to produce the final state $|\psi(t)\rangle$ within in some error tolerance ϵ.

Quantum Computing 101



"... nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

Richard Feynman (1981)

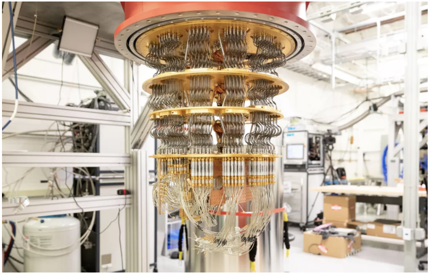
Hamiltonian Simulation Problem (original motivation for quantum computers): Given a description of the Hamiltonian $H(t)$, an evolution time t and an initial state $|\psi(0)\rangle$, to produce the final state $|\psi(t)\rangle$ within in some error tolerance ϵ.

$$\mathrm{i}\partial_t |\psi(t)\rangle = H(t) |\psi(t)\rangle , \quad |\psi(0)\rangle = |\psi_0\rangle .$$

$H(t) \equiv H,$ to simulate e^{-iHt} for H of very high dimension!.

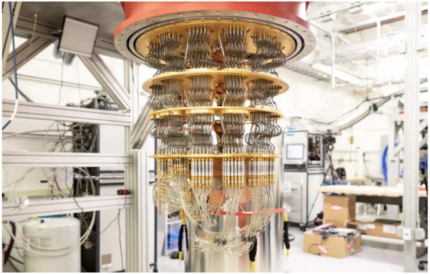
Why on a Quantum Computer?

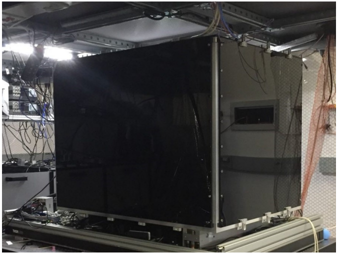
Why on a Quantum Computer?



Left: Google; Picture by Stephen Shankland (CNET). Right: Ion-trap quantum computer at Duke quantum center.

Why on a Quantum Computer?

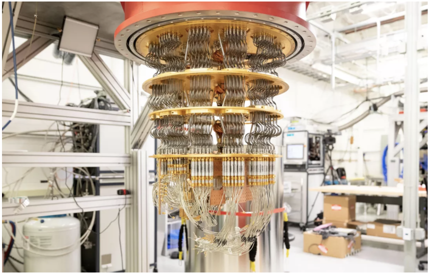


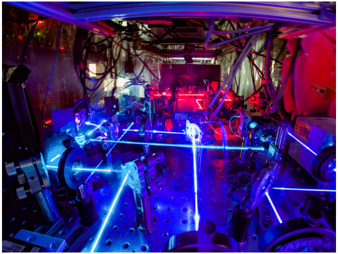


Left: Google; Picture by Stephen Shankland (CNET).

Right: Ion-trap quantum computer at Duke quantum center.

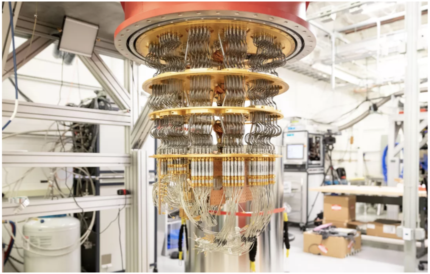
Why on a Quantum Computer?

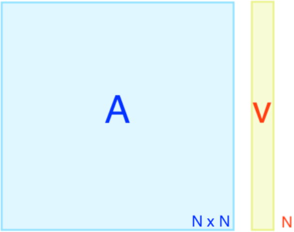




Left: Google; Picture by Stephen Shankland (CNET). *Right: Ion-trap quantum computer at Duke quantum center.*

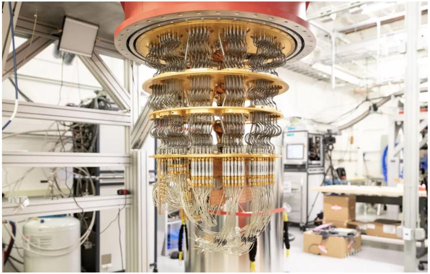
Why on a Quantum Computer?

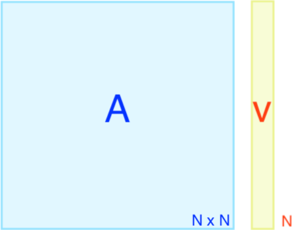




Left: Google; Picture by Stephen Shankland (CNET).

Why on a Quantum Computer?



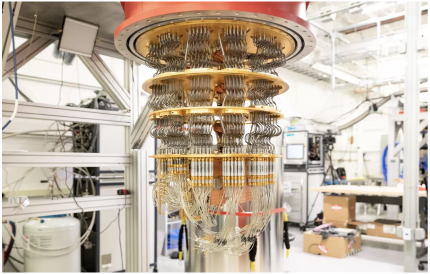


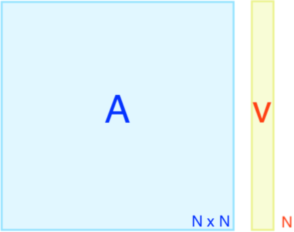
in (poly)log(N) for certain A but requiring no structure of v.

Quantum Advantage:
Quantum computers can give potential exponential speed ups.

Left: Google; Picture by Stephen Shankland (CNET). *Right: for fault-tolerant quantum computers.*

Why on a Quantum Computer?





in (poly)log(N) for certain A but requiring no structure of v.

Quantum Advantage:
Quantum computers can give potential exponential speed ups.

Potential Applications: numerical algebra, numerical differential equations, and many more scientific computing topics

Left: Google; Picture by Stephen Shankland (CNET). *Right: for fault-tolerant quantum computers.*

Part 1.2: How? Some **Basics** of Quantum Computations

Basic QC Glossary

Basic QC Glossary

- Quantum State Space:
  In quantum mechanics, the (quantum) state of a physical system is represented by a normalized vector in a Hilbert space, denoted as $\mathcal{H}$: a complex vector space with an inner product.

Basic QC Glossary

- Quantum State Space:
  In quantum mechanics, the (quantum) state of a physical system is represented by a normalized vector in a Hilbert space, denoted as $\mathcal{H}$: a complex vector space with an inner product.
- Braket Notations: For $\dim(\mathcal{H}) = N$,

$$|\psi\rangle := \psi = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{N-1} \end{pmatrix}, \ \langle\psi| := \psi^\dagger \text{ complex conjugate.}$$

  Inner product $\langle\psi|\phi\rangle := \langle\psi, \phi\rangle = \sum_{j\in[N]} \bar{\psi}_j \phi_j$.
  Normalized: $\langle\psi|\psi\rangle = 1$ for any ψ in the state space $\mathcal{H}$.

Basic QC Glossary

- Quantum State Space:
  In quantum mechanics, the (quantum) state of a physical system is represented by a normalized vector in a Hilbert space, denoted as $\mathcal{H}$: a complex vector space with an inner product.
- Braket Notations: For $\dim(\mathcal{H}) = N$,

$$|\psi\rangle := \psi = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{N-1} \end{pmatrix}, \ \langle\psi| := \psi^\dagger \text{ complex conjugate.}$$

  Inner product $\langle\psi|\phi\rangle := \langle\psi, \phi\rangle = \sum_{j\in[N]} \bar{\psi}_j \phi_j$.
  Normalized: $\langle\psi|\psi\rangle = 1$ for any ψ in the state space $\mathcal{H}$.
  Outer Product of two quantum states $|x\rangle$ and $|y\rangle$:

$$|x\rangle\langle y| = \begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_{N-1} \end{pmatrix} \begin{pmatrix} y_0^\dagger & y_1^\dagger & \cdots & y_{N-1}^\dagger \end{pmatrix}$$

Basic QC Glossary

- Quantum State Space:
  In quantum mechanics, the (quantum) state of a physical system is represented by a normalized vector in a Hilbert space, denoted as $\mathcal{H}$: a complex vector space with an inner product.
- Braket Notations: For $\dim(\mathcal{H}) = N$,

$$|\psi\rangle := \psi = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{N-1} \end{pmatrix}, \ \langle\psi| := \psi^\dagger \text{ complex conjugate.}$$

Inner product $\langle\psi|\phi\rangle := \langle\psi, \phi\rangle = \sum_{j\in[N]} \bar{\psi}_j \phi_j$.
Outer Product of two quantum states $|x\rangle$ and $|y\rangle$:

$$|x\rangle\langle y| = \begin{pmatrix} x_0 \\ x_1 \\ \cdots \\ x_{N-1} \end{pmatrix} \begin{pmatrix} y_0^\dagger & y_1^\dagger & \cdots & y_{N-1}^\dagger \end{pmatrix} \text{ maps } |y\rangle \text{ to } |x\rangle.$$

$\mathrm{Tr}(|x\rangle\langle y|) = \langle y|x\rangle$, $\quad |x\rangle\langle x|$ is a projection operator.

Basic QC Glossary – one qubit & superposition

- Simple example: 2 dimensional case

  Standard/ Computational Basis Vectors $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Basic QC Glossary – one qubit & superposition

- Simple example: 2 dimensional case

  Standard/ Computational Basis Vectors $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

  Superposition: A (Quantum) State

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle ,$$

  for $\alpha, \beta \in \mathbb{C}$ s.t. $|\alpha|^2 + |\beta|^2 = 1$.

Basic QC Glossary – one qubit & superposition

- Simple example: 2 dimensional case

  Standard/ Computational Basis Vectors $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

  Superposition: A (Quantum) State

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

  for $\alpha, \beta \in \mathbb{C}$ s.t. $|\alpha|^2 + |\beta|^2 = 1$. $\Rightarrow$ One Qubit (quantum bit)!

Basic QC Glossary – one qubit & superposition

- Simple example: 2 dimensional case

  Standard/ Computational Basis Vectors $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

  Superposition: A (Quantum) State of a one-qubit system

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

  for $\alpha, \beta \in \mathbb{C}$ s.t. $|\alpha|^2 + |\beta|^2 = 1$. $\Rightarrow$ One Qubit (quantum bit)! When we observe (or measure) in this basis, we "see" (get an outcome of) 0 with probability $|\alpha|^2$, and 1 with probability $|\beta|^2$.

Classical Bit

Basic QC Glossary – one qubit & superposition

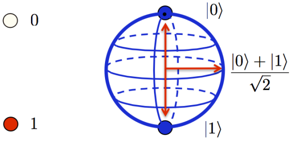
- Simple example: 2 dimensional case

  Standard/ Computational Basis Vectors $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

  Superposition: A (Quantum) State of a one-qubit system

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

  for $\alpha, \beta \in \mathbb{C}$ s.t. $|\alpha|^2 + |\beta|^2 = 1$. $\Rightarrow$ One Qubit (quantum bit)! When we observe (or measure) in this basis, we "see" (get an outcome of) 0 with probability $|\alpha|^2$, and 1 with probability $|\beta|^2$.



Classical Bit **Qubit**

Basic QC Glossary – one qubit & superposition

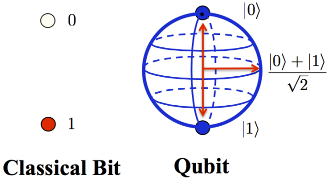
- Simple example: 2 dimensional case

  Standard/ Computational Basis Vectors $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

  Superposition: A (Quantum) State of a one-qubit system

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

  for $\alpha, \beta \in \mathbb{C}$ s.t. $|\alpha|^2 + |\beta|^2 = 1$. $\Rightarrow$ One Qubit (quantum bit)! When we observe (or measure) in this basis, we "see" (get an outcome of) 0 with probability $|\alpha|^2$, and 1 with probability $|\beta|^2$.



Geometry of a qubit: Bloch Sphere

Basic QC Glossary – one qubit & superposition

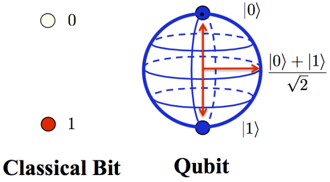
- Simple example: 2 dimensional case

Standard/ Computational Basis Vectors $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Superposition: A (Quantum) State of a one-qubit system

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

for $\alpha, \beta \in \mathbb{C}$ s.t. $|\alpha|^2 + |\beta|^2 = 1$. $\Rightarrow$ One Qubit (quantum bit)!
When we observe (or measure) in this basis, we "see" (get an outcome of) 0 with probability $|\alpha|^2$, and 1 with probability $|\beta|^2$.



Geometry of a qubit: Bloch Sphere

Wait... Something is off?

Basic QC Glossary – one-qubit state

Quantum Principle: Physical properties remain unchanged w.r.t. a global phase.

$$|\psi\rangle \to e^{i\theta} |\psi\rangle, \quad \theta \in \mathbb{R}.$$

Undistinguishable under the laws of quantum mechanism.

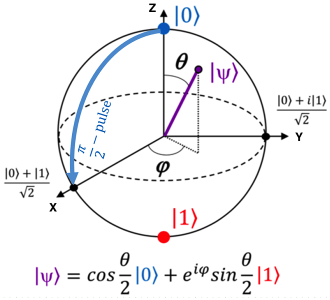
fig from QuTech.

Basic QC Glossary – one-qubit state

Quantum Principle: Physical properties remain unchanged w.r.t. a global phase.

$$|\psi\rangle \to e^{i\theta} |\psi\rangle , \quad \theta \in \mathbb{R}.$$

Undistinguishable under the laws of quantum mechanism.



$|0\rangle , |1\rangle$ also called Z basis states.

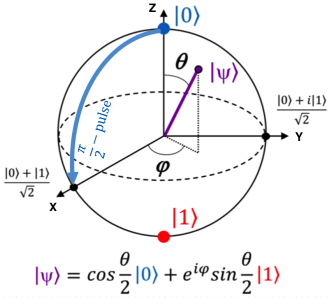
fig from QuTech.

Basic QC Glossary – one-qubit state

Quantum Principle: Physical properties remain unchanged w.r.t. a global phase.

$$|\psi\rangle \to e^{i\theta} |\psi\rangle, \quad \theta \in \mathbb{R}.$$

Undistinguishable under the laws of quantum mechanism.



$|0\rangle, |1\rangle$ also called Z basis states.

When $\theta = \pi/2$, $\phi = 0$,
$|+\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$

When $\theta = \pi/2$, $\phi = \pi$,
$|-\rangle = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$

called the X basis states.

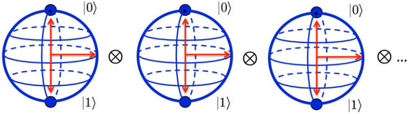
fig from QuTech.

Basic QC Glossary – multi-qubit state

- For general n-qubit system, $\mathcal{H} = \mathcal{B}^{\otimes n}$.

Basic QC Glossary – multi-qubit state

- For general n-qubit system, $\mathcal{H} = \mathcal{B}^{\otimes n}$.
- Tensor Product

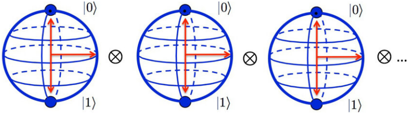


$$|x\rangle \otimes |y\rangle = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \\ x_1 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} x_0 y_0 \\ x_0 y_1 \\ x_1 y_0 \\ x_1 y_1 \end{pmatrix}$$

E.g., $|00\rangle := |0\rangle \otimes |0\rangle =$, $|01\rangle := |0\rangle \otimes |1\rangle =$,
$|10\rangle := |1\rangle \otimes |0\rangle =$, $|11\rangle := |1\rangle \otimes |1\rangle =$.

Basic QC Glossary – multi-qubit state

- For general n-qubit system, $\mathcal{H} = \mathcal{B}^{\otimes n}$.
- Tensor Product



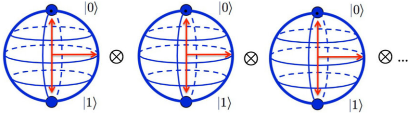
$$|x\rangle \otimes |y\rangle = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \\ x_1 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} x_0 y_0 \\ x_0 y_1 \\ x_1 y_0 \\ x_1 y_1 \end{pmatrix}$$

E.g., $|00\rangle := |0\rangle \otimes |0\rangle = (1,0,0,0)^T$, $|01\rangle := |0\rangle \otimes |1\rangle = (0,1,0,0)^T$,
$|10\rangle := |1\rangle \otimes |0\rangle = (0,0,1,0)^T$, $|11\rangle := |1\rangle \otimes |1\rangle = (0,0,0,1)^T$.

Tensor product is non-commutative!

Basic QC Glossary – multi-qubit state

- For general n-qubit system, $\mathcal{H} = \mathcal{B}^{\otimes n}$.
- Tensor Product



$$|x\rangle \otimes |y\rangle = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \\ x_1 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} x_0 y_0 \\ x_0 y_1 \\ x_1 y_0 \\ x_1 y_1 \end{pmatrix}$$

E.g., $|00\rangle := |0\rangle \otimes |0\rangle = (1,0,0,0)^T$, $|01\rangle := |0\rangle \otimes |1\rangle = (0,1,0,0)^T$, $|10\rangle := |1\rangle \otimes |0\rangle = (0,0,1,0)^T$, $|11\rangle := |1\rangle \otimes |1\rangle = (0,0,0,1)^T$.

Tensor product is non-commutative!

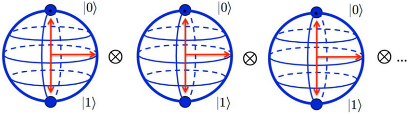
In quantum braket notation,

$$\begin{aligned} |x\rangle \otimes |y\rangle :=& (x_0 |0\rangle + x_1 |1\rangle) \otimes (y_0 |0\rangle + y_1 |1\rangle) \\ =& x_0 y_0 |00\rangle + x_0 y_1 |01\rangle + x_1 y_0 |10\rangle + x_1 y_1 |11\rangle . \end{aligned}$$

What is the relationship with N and n?

Basic QC Glossary – multi-qubit state

- For general n-qubit system, $\mathcal{H} = \mathcal{B}^{\otimes n}$.
- Tensor Product



$$|x\rangle \otimes |y\rangle = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \\ x_1 \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} x_0 y_0 \\ x_0 y_1 \\ x_1 y_0 \\ x_1 y_1 \end{pmatrix}$$

E.g., $|00\rangle := |0\rangle \otimes |0\rangle = (1,0,0,0)^T$, $|01\rangle := |0\rangle \otimes |1\rangle = (0,1,0,0)^T$,
$|10\rangle := |1\rangle \otimes |0\rangle = (0,0,1,0)^T$, $|11\rangle := |1\rangle \otimes |1\rangle = (0,0,0,1)^T$.

Tensor product is non-commutative!

In quantum braket notation,

$$\begin{aligned} |x\rangle \otimes |y\rangle :=& (x_0 |0\rangle + x_1 |1\rangle) \otimes (y_0 |0\rangle + y_1 |1\rangle) \\ =& x_0 y_0 |00\rangle + x_0 y_1 |01\rangle + x_1 y_0 |10\rangle + x_1 y_1 |11\rangle . \end{aligned}$$

What is the relationship with N and n? $\boxed{N = 2^n}$!

Basic QC Glossary – multi-qubit state & entanglement

- An n-qubit state is called a product state, if it can be represented as the tensor product of one-qubit states $|\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_n\rangle$. E.g., $|00\rangle$, $\frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)$

Basic QC Glossary – multi-qubit state & entanglement

- An n-qubit state is called a product state, if it can be represented as the tensor product of one-qubit states $|\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_n\rangle$. E.g., $|00\rangle$, $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = |+\rangle \otimes |+\rangle$
- Are all n-qubit states are product states?

Basic QC Glossary – multi-qubit state & entanglement

- An n-qubit state is called a product state, if it can be represented as the tensor product of one-qubit states $|\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_n\rangle$.
  E.g., $|00\rangle$, $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = |+\rangle \otimes |+\rangle$
- Are all n-qubit states are product states? No! Entangled States
  E.g., $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ Bell State (EPR pair)

Basic QC Glossary – multi-qubit state & entanglement

- An n-qubit state is called a product state, if it can be represented as the tensor product of one-qubit states $|\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_n\rangle$.
  E.g., $|00\rangle$, $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = |+\rangle \otimes |+\rangle$
- Are all n-qubit states are product states? No! Entangled States
  E.g., $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ Bell State (EPR pair)
  Proof: Suppose $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle)$
  $ac = bd = 1/\sqrt{2}, \quad ad = bc = 0$. Impossible. $\square$

Two important quantum features:

Superposition and Entanglement

Basic QC Glossary – operators: quantum gates

Are quantum state allowed to change over time?

Basic QC Glossary – operators: quantum gates

Are quantum state allowed to change over time? Yes!

- Quantum Gates: unitary operators acting over the state space $\mathcal{H}$. A gate acting on n qubits is represented by $2^n \times 2^n$ unitary matrix (denote as U).

$$|\psi\rangle \;-\!\!\!-\!\boxed{U}\!-\!\!\!-\; U\,|\psi\rangle$$

Basic QC Glossary – operators: quantum gates

Are quantum state allowed to change over time? Yes!

- Quantum Gates: unitary operators acting over the state space $\mathcal{H}$. A gate acting on n qubits is represented by $2^n \times 2^n$ unitary matrix (denote as U).

$$|\psi\rangle \; — \boxed{U} — \; U|\psi\rangle$$

- Properties:
  - Quantum gates preserve the norm.

    Proof: $(U|\psi\rangle)^\dagger U|\psi\rangle = \langle\psi| U^\dagger U |\psi\rangle = \langle\psi|\psi\rangle = 1$, for $|\psi\rangle \in \mathcal{H}$.

Basic QC Glossary – operators: quantum gates

Are quantum state allowed to change over time? Yes!

- Quantum Gates: unitary operators acting over the state space $\mathcal{H}$. A gate acting on n qubits is represented by $2^n \times 2^n$ unitary matrix (denote as U).

$$|\psi\rangle \;\text{—}\boxed{U}\text{—}\; U|\psi\rangle$$

- Properties:
  - Quantum gates preserve the norm.

    Proof: $(U|\psi\rangle)^\dagger U|\psi\rangle = \langle\psi| U^\dagger U|\psi\rangle = \langle\psi|\psi\rangle = 1$, for $|\psi\rangle \in \mathcal{H}$.
  - Quantum gates preserve angle between two quantum states.

Basic QC Glossary – operators: quantum gates

Are quantum state allowed to change over time? Yes!

- Quantum Gates: unitary operators acting over the state space $\mathcal{H}$. A gate acting on n qubits is represented by $2^n \times 2^n$ unitary matrix (denote as U).

$$|\psi\rangle \; - \boxed{U} - \; U|\psi\rangle$$

- Properties:
  - Quantum gates preserve the norm.

    Proof: $(U|\psi\rangle)^\dagger U|\psi\rangle = \langle\psi| U^\dagger U |\psi\rangle = \langle\psi|\psi\rangle = 1$, for $|\psi\rangle \in \mathcal{H}$.
  - Quantum gates preserve angle between two quantum states.
  - Quantum gates are invertible (reversible).

Basic QC Glossary – operators: quantum gates

Are quantum state allowed to change over time? Yes!

- Quantum Gates: unitary operators acting over the state space $\mathcal{H}$. A gate acting on n qubits is represented by $2^n \times 2^n$ unitary matrix (denote as U).

$$|\psi\rangle \;—\boxed{U}—\; U|\psi\rangle$$

- Properties:
    - Quantum gates preserve the norm.
      Proof: $(U|\psi\rangle)^\dagger U|\psi\rangle = \langle\psi| U^\dagger U |\psi\rangle = \langle\psi|\psi\rangle = 1$, for $|\psi\rangle \in \mathcal{H}$.
    - Quantum gates preserve angle between two quantum states.
    - Quantum gates are invertible (reversible).
- Examples – commonly used single-qubit gates:

Hadamard Gate $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $H^\dagger = H^{-1} = H$

$|0\rangle \;—\boxed{H}—\;$? $\quad |1\rangle \;—\boxed{H}—\;$?

Basic QC Glossary – operators: quantum gates

Are quantum state allowed to change over time? Yes!

- Quantum Gates: unitary operators acting over the state space $\mathcal{H}$. A gate acting on n qubits is represented by $2^n \times 2^n$ unitary matrix (denote as U).

$$|\psi\rangle \;-\!\boxed{U}\!-\; U|\psi\rangle$$

- Properties:
  - Quantum gates preserve the norm.

    Proof: $(U|\psi\rangle)^\dagger U|\psi\rangle = \langle\psi| U^\dagger U |\psi\rangle = \langle\psi|\psi\rangle = 1$, for $|\psi\rangle \in \mathcal{H}$.
  - Quantum gates preserve angle between two quantum states.
  - Quantum gates are invertible (reversible).
- Examples – commonly used single-qubit gates:

Hadamard Gate $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $H^\dagger = H^{-1} = H$

$$|0\rangle \;-\!\boxed{H}\!-\; |+\rangle \qquad |1\rangle \;-\!\boxed{H}\!-\; |-\rangle \quad H = |+\rangle\langle 0| + |-\rangle\langle 1|$$

Basic QC Glossary – common one-qubit gates cont'd

- Pauli matrices $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$
  (also denote as σ_x, σ_y, σ_z.) $Y = iXZ$

Basic QC Glossary – common one-qubit gates cont'd

- Pauli matrices $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$
  (also denote as σ_x, σ_y, σ_z.) $Y = iXZ$
  $X : |0\rangle \leftrightarrow |1\rangle$ bit flip; $Z : |x\rangle \rightarrow (-1)^x |x\rangle$, $x = 0, 1$ phase-flip.

Basic QC Glossary – common one-qubit gates cont'd

- Pauli matrices $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$
  (also denote as σ_x, σ_y, σ_z.) $Y = iXZ$
  $X : |0\rangle \leftrightarrow |1\rangle$ bit flip; $Z : |x\rangle \to (-1)^x |x\rangle$, $x = 0, 1$ phase-flip.
  Multi-qubit Paulis: tensors of single qubit Paulis.
  Properties:

Basic QC Glossary – common one-qubit gates cont'd

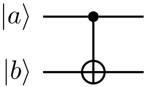
- Pauli matrices $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$
  (also denote as σ_x, σ_y, σ_z.) $Y = iXZ$
  $X : |0\rangle \leftrightarrow |1\rangle$ bit flip; $Z : |x\rangle \to (-1)^x |x\rangle$, $x = 0, 1$ phase-flip.
  Multi-qubit Paulis: tensors of single qubit Paulis.
  Properties:
  - Their inverses are themselves. (Hermitian + Unitary)
  - X/Y/Z basis vectors are eigenvectors of X, Y, Z, respectively.
  - They anti-commute.
  - (many-body) Hamiltonian (Hermitian matrices) can be written as linear combinations of (n-qubit) Paulis.

Basic QC Glossary – common one-qubit gates cont'd

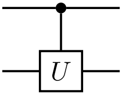
- Pauli matrices $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$
  (also denote as σ_x, σ_y, σ_z.) $Y = iXZ$
  $X : |0\rangle \leftrightarrow |1\rangle$ bit flip; $Z : |x\rangle \to (-1)^x |x\rangle$, $x = 0, 1$ phase-flip.
  Multi-qubit Paulis: tensors of single qubit Paulis.
  Properties:
  - Their inverses are themselves. (Hermitian + Unitary)
  - X/Y/Z basis vectors are eigenvectors of X, Y, Z, respectively.
  - They anti-commute.
  - (many-body) Hamiltonian (Hermitian matrices) can be written as linear combinations of (n-qubit) Paulis.
- Phase-shift Gate: $P(\phi) = P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}$
  $Z = P(\pi)$, $S = P(\pi/2) = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$, $T = P(\pi/4) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$

Common two-qubit gates

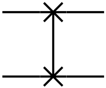
- CNOT (controlled-NOT)



- controlled-U gate

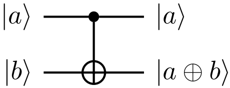


- SWAP

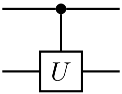


Common two-qubit gates

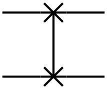
- CNOT (controlled-NOT)



- controlled-U gate

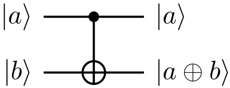


- SWAP

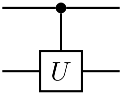


Common two-qubit gates

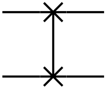
- CNOT (controlled-NOT)



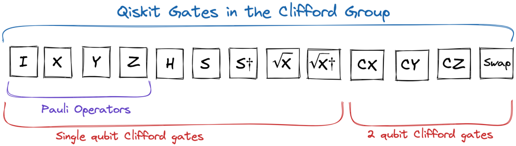
- controlled-U gate



- SWAP

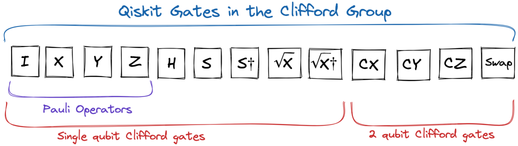


Clifford gates: elements of Clifford group
$\mathbf{C}_n = \{V \in U_{2^n} \mid V\mathbf{P}_n V^\dagger = \mathbf{P}_n\}$. Here P_n is the n-qubit Pauli group.
Generators: $\{$H, S, CNOT$\}$.



Non-Clifford gates:
T, Toffoli (CCNOT)

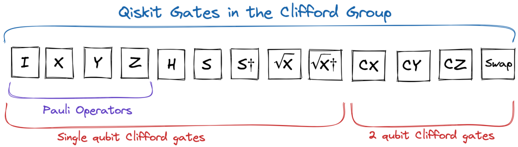
Clifford gates: elements of Clifford group
$\mathbf{C}_n = \{V \in U_{2^n} \mid V\mathbf{P}_nV^\dagger = \mathbf{P}_n\}$. Here P_n is the n-qubit Pauli group.
Generators: $\{$H, S, CNOT$\}$.



Non-Clifford gates:
T, Toffoli (CCNOT)

3-qubit gates? 4-qubit gates? General n-qubit gates? tons of gates to remember??

Clifford gates: elements of Clifford group
$\mathbf{C}_n = \{V \in U_{2^n} \mid V\mathbf{P}_n V^\dagger = \mathbf{P}_n\}$. Here P_n is the n-qubit Pauli group.
Generators: {H, S, CNOT}.



Non-Clifford gates:
T, Toffoli (CCNOT)

3-qubit gates? 4-qubit gates? General n-qubit gates? tons of gates to remember??

Upshot: Universality!

A set of quantum gates is called universal, if composing gates from it can approximate any quantum gate to any desired precision.
Some examples of universal gate sets are:

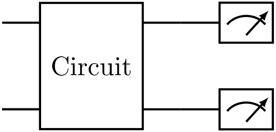
- {CNOT, all single-qubit gates}
- {CNOT, H, T}
- {Toffoli, H}

Basic QC Glossary – Measurements

- Measurement:

Basic QC Glossary – Measurements

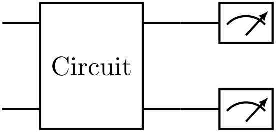
- Measurement:
  2-qubit example $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$



We will observe

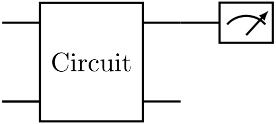
Basic QC Glossary – Measurements

- Measurement:
  2-qubit example $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$



We will observe $\begin{cases} 00 \text{ with prob } |\alpha_{00}|^2 \\ 01 \text{ with prob } |\alpha_{01}|^2 \\ 10 \text{ with prob } |\alpha_{10}|^2 \\ 11 \text{ with prob } |\alpha_{11}|^2 \end{cases}$

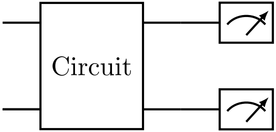
- Partial Measurement:



Observe

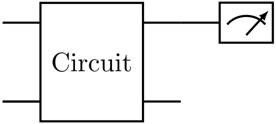
Basic QC Glossary – Measurements

- Measurement:
  2-qubit example $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$



We will observe $\begin{cases} 00 \text{ with prob } |\alpha_{00}|^2 \\ 01 \text{ with prob } |\alpha_{01}|^2 \\ 10 \text{ with prob } |\alpha_{10}|^2 \\ 11 \text{ with prob } |\alpha_{11}|^2 \end{cases}$

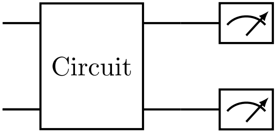
- Partial Measurement:



Observe $\begin{cases} 0 \text{ with prob } |\alpha_{00}|^2 + |\alpha_{01}|^2 \\ 1 \text{ with prob } |\alpha_{10}|^2 + |\alpha_{11}|^2 \end{cases}$

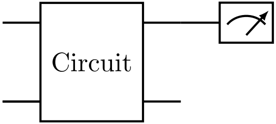
Basic QC Glossary – Measurements

- Measurement:
  2-qubit example $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$



We will observe $\begin{cases} 00 \text{ with prob } |\alpha_{00}|^2 \\ 01 \text{ with prob } |\alpha_{01}|^2 \\ 10 \text{ with prob } |\alpha_{10}|^2 \\ 11 \text{ with prob } |\alpha_{11}|^2 \end{cases}$

- Partial Measurement:



Observe $\begin{cases} 0 \text{ with prob } |\alpha_{00}|^2 + |\alpha_{01}|^2 \\ 1 \text{ with prob } |\alpha_{10}|^2 + |\alpha_{11}|^2 \end{cases}$

If we observe 0, the joint state after the measurement becomes

$$\frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}} = |0\rangle \otimes \frac{\alpha_{00}|0\rangle + \alpha_{01}|1\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}.$$ "Unentangled"

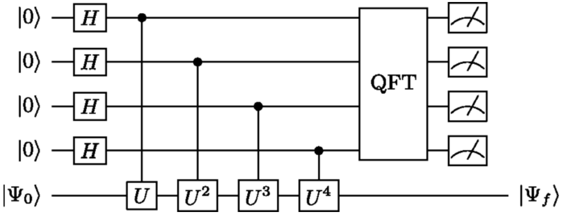
Wave function collapse after measurement

Basic QC Glossary – quantum circuits

Quantum algorithms (QA) are represented by quantum circuits.

Basic QC Glossary – quantum circuits

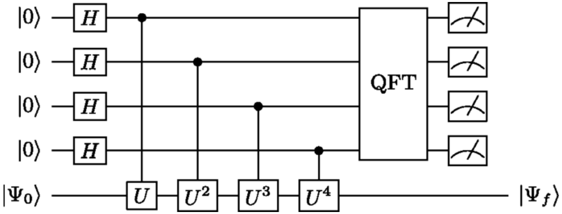
Quantum algorithms (QA) are represented by quantum circuits.



Complexity of a QA: Gate complexity, Query complexity (oracle)

Basic QC Glossary – quantum circuits

Quantum algorithms (QA) are represented by quantum circuits.

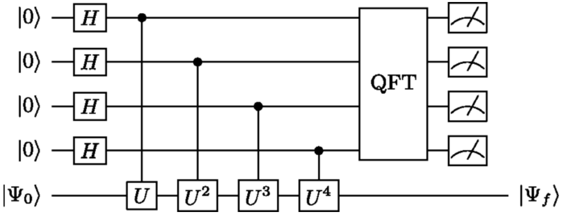


Complexity of a QA: Gate complexity, Query complexity (oracle)

Question: Relationship of QA v.s. Classical algorithms?

Basic QC Glossary – quantum circuits

Quantum algorithms (QA) are represented by quantum circuits.



Complexity of a QA: Gate complexity, Query complexity (oracle)

Question: Relationship of QA v.s. Classical algorithms?

- Is QC at least as powerful as classical computing?
- Is there always an exponential (superpolynomial) quantum advantage?

Basic QC Glossary – QC vs CC

Question 1: Is QC at least as powerful as classical computing?

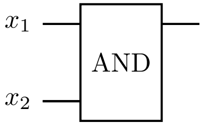
Basic QC Glossary – QC vs CC

Question 1: Is QC at least as powerful as classical computing? Yes!
Classical arithmetic operations can be performed quantumly.

Basic QC Glossary – QC vs CC

Question 1: Is QC at least as powerful as classical computing? Yes!
Classical arithmetic operations can be performed quantumly.

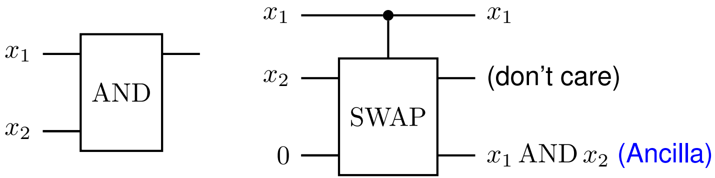
Really? Reversibility?



Basic QC Glossary – QC vs CC

Question 1: Is QC at least as powerful as classical computing? Yes!
Classical arithmetic operations can be performed quantumly.

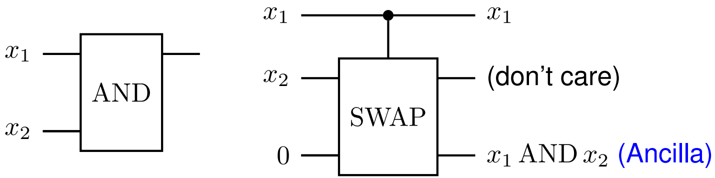
Really? Reversibility?



Basic QC Glossary – QC vs CC

Question 1: Is QC at least as powerful as classical computing? Yes!
Classical arithmetic operations can be performed quantumly.

Really? Reversibility?

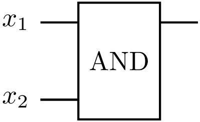


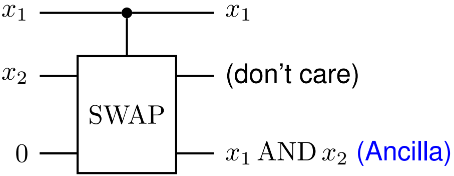
What about Probabilistic Computing? Success prob $\geq$ (say 0.99)

Basic QC Glossary – QC vs CC

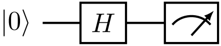
Question 1: Is QC at least as powerful as classical computing? Yes!
Classical arithmetic operations can be performed quantumly.

Really? Reversibility?





What about Probabilistic Computing? Success prob $\geq$ (say 0.99)

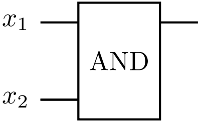


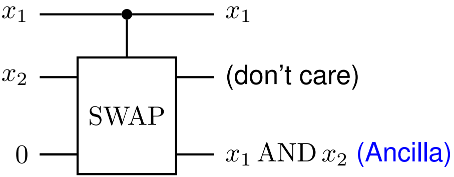
Question 2: Always exponential quantum speedup (b/c $2^n = N$)?

Basic QC Glossary – QC vs CC

Question 1: Is QC at least as powerful as classical computing? Yes!
Classical arithmetic operations can be performed quantumly.

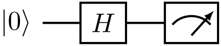
Really? Reversibility?





What about Probabilistic Computing? Success prob $\geq$ (say 0.99)

$\boxed{\text{CoinFlip}}$



Question 2: Always exponential quantum speedup (b/c $2^n = N$)?
No!! Restrictions:

- Unitary + Measurement (Needs structure of tasks!)
- No cloning theorem

Basic QC Glossary: No-cloning Theorem

There is no quantum circuit that clones an arbitrary quantum state!

Basic QC Glossary: No-cloning Theorem

There is no quantum circuit that clones an arbitrary quantum state!

- No-cloning Theorem (simple ver.): There is no unitary operation that can enact the evolution $|\psi\rangle\,|s\rangle \to |\psi\rangle\,|\psi\rangle$ for all states $|\psi\rangle$.

Basic QC Glossary: No-cloning Theorem

There is no quantum circuit that clones an arbitrary quantum state!

- No-cloning Theorem (simple ver.): There is no unitary operation that can enact the evolution $|\psi\rangle|s\rangle \to |\psi\rangle|\psi\rangle$ for all states $|\psi\rangle$.
  Proof: Suppose there exists U s.t. $U|\psi\rangle|s\rangle = |\psi\rangle|\psi\rangle$ for all $|\psi\rangle$.
  $\Rightarrow U|x_1\rangle|s\rangle = |x_1\rangle|x_1\rangle, \quad U|x_2\rangle|s\rangle = |x_2\rangle|x_1\rangle$

Basic QC Glossary: No-cloning Theorem

There is no quantum circuit that clones an arbitrary quantum state!

- No-cloning Theorem (simple ver.): There is no unitary operation that can enact the evolution $|\psi\rangle|s\rangle \to |\psi\rangle|\psi\rangle$ for all states $|\psi\rangle$.

  Proof: Suppose there exists U s.t. $U|\psi\rangle|s\rangle = |\psi\rangle|\psi\rangle$ for all $|\psi\rangle$.
  $\Rightarrow U|x_1\rangle|s\rangle = |x_1\rangle|x_1\rangle\,, \quad U|x_2\rangle|s\rangle = |x_2\rangle|x_1\rangle$
  $\Rightarrow \langle x_1|x_2\rangle = \langle x_1|x_2\rangle^2$ (taking the inner product)

Basic QC Glossary: No-cloning Theorem

There is no quantum circuit that clones an arbitrary quantum state!

- No-cloning Theorem (simple ver.): There is no unitary operation that can enact the evolution $|\psi\rangle |s\rangle \to |\psi\rangle |\psi\rangle$ for all states $|\psi\rangle$.

  Proof: Suppose there exists U s.t. $U|\psi\rangle |s\rangle = |\psi\rangle |\psi\rangle$ for all $|\psi\rangle$.
  $\Rightarrow U|x_1\rangle |s\rangle = |x_1\rangle |x_1\rangle, \quad U|x_2\rangle |s\rangle = |x_2\rangle |x_1\rangle$
  $\Rightarrow \langle x_1|x_2\rangle = \langle x_1|x_2\rangle^2$ (taking the inner product)
  $\Rightarrow \langle x_1|x_2\rangle = 0$ or 1. Contradiction! $\square$

Basic QC Glossary: No-cloning Theorem

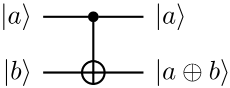
There is no quantum circuit that clones an arbitrary quantum state!

- No-cloning Theorem (simple ver.): There is no unitary operation that can enact the evolution $|\psi\rangle |s\rangle \to |\psi\rangle |\psi\rangle$ for all states $|\psi\rangle$.
- Consequence: Iterative-type algorithms for scientific computing tasks are difficult to implement efficiently.

Basic QC Glossary: No-cloning Theorem

There is no quantum circuit that clones an arbitrary quantum state!

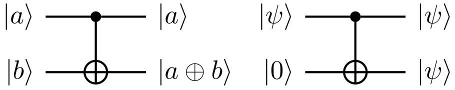
- No-cloning Theorem (simple ver.): There is no unitary operation that can enact the evolution $|\psi\rangle |s\rangle \to |\psi\rangle |\psi\rangle$ for all states $|\psi\rangle$.
- Consequence: Iterative-type algorithms for scientific computing tasks are difficult to implement efficiently.
- Wait... Something is weird? CNOT



Basic QC Glossary: No-cloning Theorem

There is no quantum circuit that clones an arbitrary quantum state!

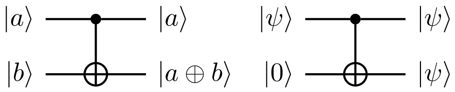
- No-cloning Theorem (simple ver.): There is no unitary operation that can enact the evolution $|\psi\rangle |s\rangle \to |\psi\rangle |\psi\rangle$ for all states $|\psi\rangle$.
- Consequence: Iterative-type algorithms for scientific computing tasks are difficult to implement efficiently.
- Wait... Something is weird? CNOT



Basic QC Glossary: No-cloning Theorem

There is no quantum circuit that clones an arbitrary quantum state!

- No-cloning Theorem (simple ver.): There is no unitary operation that can enact the evolution $|\psi\rangle\,|s\rangle \to |\psi\rangle\,|\psi\rangle$ for all states $|\psi\rangle$.
- Consequence: Iterative-type algorithms for scientific computing tasks are difficult to implement efficiently.
- Wait... Something is weird? CNOT

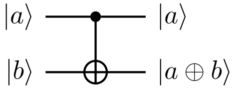


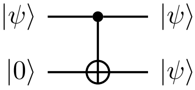
Copying of *classical* info. No contrad.!

Basic QC Glossary: No-cloning Theorem

There is no quantum circuit that clones an arbitrary quantum state!

- No-cloning Theorem (simple ver.): There is no unitary operation that can enact the evolution $|\psi\rangle\,|s\rangle \to |\psi\rangle\,|\psi\rangle$ for all states $|\psi\rangle$.
- Consequence: Iterative-type algorithms for scientific computing tasks are difficult to implement efficiently.
- Wait... Something is weird? CNOT





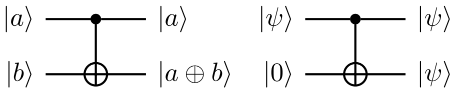
Copying of *classical* info. No contrad.!

- Another important remark: If we know how to prepare $|\psi\rangle$ (from $|s\rangle$), i.e. $|\psi\rangle = U\,|s\rangle$ for a known unitary U. Then

Basic QC Glossary: No-cloning Theorem

There is no quantum circuit that clones an arbitrary quantum state!

- No-cloning Theorem (simple ver.): There is no unitary operation that can enact the evolution $|\psi\rangle\,|s\rangle \to |\psi\rangle\,|\psi\rangle$ for all states $|\psi\rangle$.
- Consequence: Iterative-type algorithms for scientific computing tasks are difficult to implement efficiently.
- Wait... Something is weird? CNOT



Copying of *classical* info. No contrad.!

- Another important remark: If we know how to prepare $|\psi\rangle$ (from $|s\rangle$), i.e. $|\psi\rangle = U\,|s\rangle$ for a known unitary U. Then

$$(I \otimes U)\,|\psi\rangle\,|s\rangle = |\psi\rangle\,|\psi\rangle\,.$$

Basic QC Glossary: Exponential Quantum Advantage

Exponential Quantum Advantage (EQA) (often this is also used to refer superpolynomial speedup)

Criteria to claim EQA:

Basic QC Glossary: Exponential Quantum Advantage

Exponential Quantum Advantage (EQA) (often this is also used to refer superpolynomial speedup)

Criteria to claim EQA:

- There is a QA with quantum complexity $\leq \operatorname{polylog} N$.

Basic QC Glossary: Exponential Quantum Advantage

Exponential Quantum Advantage (EQA) (often this is also used to refer superpolynomial speedup)

Criteria to claim EQA:

- There is a QA with quantum complexity $\leq \operatorname{polylog} N$.
- $\begin{cases} \text{(A) Best-known Classical Alg. has complexity } \geq e^{\operatorname{polylog} N} \\ \end{cases}$

Basic QC Glossary: Exponential Quantum Advantage

Exponential Quantum Advantage (EQA) (often this is also used to refer superpolynomial speedup)

Criteria to claim EQA:

- There is a QA with quantum complexity $\leq \operatorname{polylog} N$.
- $$\begin{cases} \text{(A) Best-known Classical Alg. has complexity } \geq e^{\operatorname{polylog} N} \\ \text{(B) Show that the task is BQP-complete} \\ (\text{Any Classical Alg. under reasonable complexity conjectures}) \end{cases}$$

Basic QC Glossary: Exponential Quantum Advantage

Exponential Quantum Advantage (EQA) (often this is also used to refer superpolynomial speedup)

Criteria to claim EQA:

- There is a QA with quantum complexity $\leq \operatorname{polylog} N$.
- $$\begin{cases} \text{(A) Best-known Classical Alg. has complexity } \geq e^{\operatorname{polylog} N} \\ \text{(B) Show that the task is BQP-complete} \\ (\text{Any Classical Alg. under reasonable complexity conjectures}) \end{cases}$$

Examples of Tasks with EQA:

- Factoring $\Rightarrow$ Shor's Algorithm (A)
- Invert a large sparse linear system $\Rightarrow$ HHL Algorithm (A)(B)
- Hamiltonian Simulation (B)

Summary of Part 1

- Motivation: first principle, potential EQA
- Quantum State
- Quantum Gates / Circuits
- Measurement
- QA v.s. CA; no-cloning; EQA

Part 2: Block-encoding and Hamiltonian Simulation

Hamiltonian Simulation

Hamiltonian Simulation Problem: Given a description of the Hamiltonian $H(t)$, an evolution time t and an initial state $|\psi(0)\rangle$, to produce the final state $|\psi(t)\rangle$ within in some error tolerance ϵ.

Hamiltonian Simulation

Hamiltonian Simulation Problem: Given a description of the Hamiltonian $H(t)$, an evolution time t and an initial state $|\psi(0)\rangle$, to produce the final state $|\psi(t)\rangle$ within in some error tolerance ϵ.

Time-independent: $H(t) \equiv H$ is a $2^n \times 2^n$ matrix

$$\mathrm{i}\partial_t |\psi(t)\rangle = H |\psi(t)\rangle , \quad |\psi(0)\rangle = |\psi_0\rangle .$$

Hamiltonian Simulation

Hamiltonian Simulation Problem: Given a description of the Hamiltonian $H(t)$, an evolution time t and an initial state $|\psi(0)\rangle$, to produce the final state $|\psi(t)\rangle$ within in some error tolerance ϵ.

Time-independent: $H(t) \equiv H$ is a $2^n \times 2^n$ matrix

$$\mathrm{i}\partial_t |\psi(t)\rangle = H |\psi(t)\rangle, \quad |\psi(0)\rangle = |\psi_0\rangle.$$

$$\left\| \mathcal{U}_{\mathsf{app}} |\psi_0\rangle - e^{-\mathrm{i}Ht} |\psi_0\rangle \right\|$$

Hamiltonian Simulation

Hamiltonian Simulation Problem: Given a description of the Hamiltonian $H(t)$, an evolution time t and an initial state $|\psi(0)\rangle$, to produce the final state $|\psi(t)\rangle$ within in some error tolerance ϵ.

Time-independent: $H(t) \equiv H$ is a $2^n \times 2^n$ matrix

$$\mathrm{i}\partial_t |\psi(t)\rangle = H |\psi(t)\rangle , \quad |\psi(0)\rangle = |\psi_0\rangle .$$

$$\left\| \mathcal{U}_{\text{app}} |\psi_0\rangle - e^{-\mathrm{i}Ht} |\psi_0\rangle \right\| \leq \left\| \mathcal{U}_{\text{app}} - e^{-\mathrm{i}Ht} \right\| \leq \epsilon.$$

Hamiltonian Simulation

Hamiltonian Simulation Problem: Given a description of the Hamiltonian $H(t)$, an evolution time t and an initial state $|\psi(0)\rangle$, to produce the final state $|\psi(t)\rangle$ within in some error tolerance ϵ.

Time-independent: $H(t) \equiv H$ is a $2^n \times 2^n$ matrix

$$\mathrm{i}\partial_t |\psi(t)\rangle = H |\psi(t)\rangle, \quad |\psi(0)\rangle = |\psi_0\rangle.$$

$$\left\| \mathcal{U}_{\text{app}} |\psi_0\rangle - e^{-\mathrm{i}Ht} |\psi_0\rangle \right\| \leq \boxed{\left\| \mathcal{U}_{\text{app}} - e^{-\mathrm{i}Ht} \right\| \leq \epsilon}.$$

Hamiltonian Simulation

Hamiltonian Simulation Problem: Given a description of the Hamiltonian $H(t)$, an evolution time t and an initial state $|\psi(0)\rangle$, to produce the final state $|\psi(t)\rangle$ within in some error tolerance ϵ.

Time-independent: $H(t) \equiv H$ is a $2^n \times 2^n$ matrix

$$\mathrm{i}\partial_t |\psi(t)\rangle = H |\psi(t)\rangle, \quad |\psi(0)\rangle = |\psi_0\rangle.$$

$$\left\| \mathcal{U}_{\text{app}} |\psi_0\rangle - e^{-\mathrm{i}Ht} |\psi_0\rangle \right\| \leq \boxed{\left\| \mathcal{U}_{\text{app}} - e^{-\mathrm{i}Ht} \right\| \leq \epsilon}.$$

Examples of H: many-body Hamiltonian

$$H = \sum_{E \in S \subset \{I,X,Y,Z\}^{\otimes n}} \lambda_E E,$$

k-local Hamiltonian (TFIM, Heisenberg models, etc), etc.

Hamiltonian Simulation

Hamiltonian Simulation Problem: Given a description of the Hamiltonian $H(t)$, an evolution time t and an initial state $|\psi(0)\rangle$, to produce the final state $|\psi(t)\rangle$ within in some error tolerance ϵ.

Time-independent: $H(t) \equiv H$ is a $2^n \times 2^n$ matrix

$$\mathrm{i}\partial_t |\psi(t)\rangle = H |\psi(t)\rangle , \quad |\psi(0)\rangle = |\psi_0\rangle .$$

$$\left\| \mathcal{U}_{\text{app}} |\psi_0\rangle - e^{-\mathrm{i}Ht} |\psi_0\rangle \right\| \leq \boxed{\left\| \mathcal{U}_{\text{app}} - e^{-\mathrm{i}Ht} \right\| \leq \epsilon}.$$

Examples of H: many-body Hamiltonian

$$H = \sum_{E \in S \subset \{I,X,Y,Z\}^{\otimes n}} \lambda_E E,$$

k-local Hamiltonian (TFIM, Heisenberg models, etc), etc.

No-fast-forwarding Theorem *(informal)*: Simulating Hamiltonian dynamics for time t requires complexity $\Omega(t)$.

Hamiltonian Simulation Algorithms

- Trotterization (= Product Formulae = Time/Operator Splitting)
  1st-order Trotter formula (Lie-Trotter) for $H = H_1 + H_2$

$$e^{-\mathrm{i}Ht} \approx \left(e^{-\mathrm{i}H_2 t/L} e^{-\mathrm{i}H_1 t/L}\right)^L$$

  Cost/Complexity?

Hamiltonian Simulation Algorithms

- Trotterization (= Product Formulae = Time/Operator Splitting)
  1st-order Trotter formula (Lie-Trotter) for $H = H_1 + H_2$
  $$e^{-\mathrm{i}Ht} \approx \left(e^{-\mathrm{i}H_2 t/L} e^{-\mathrm{i}H_1 t/L}\right)^L$$
  Cost/Complexity? error estimate and circuit implementation

Hamiltonian Simulation Algorithms

- Trotterization (= Product Formulae = Time/Operator Splitting)
  1st-order Trotter formula (Lie-Trotter) for $H = H_1 + H_2$

$$e^{-\mathrm{i}Ht} \approx \left(e^{-\mathrm{i}H_2 t/L} e^{-\mathrm{i}H_1 t/L}\right)^L$$

  Cost/Complexity? error estimate and circuit implementation

$$e^{-\mathrm{i}Ht} = \left(e^{-\mathrm{i}H_2 t/L} e^{-\mathrm{i}H_1 t/L}\right)^L + \mathcal{O}\left(\|[H_1, H_2]\| t^2 / L\right)$$

  The number of Trotter steps $L = \mathcal{O}\left(\|[H_1, H_2]\| t^2 / \epsilon\right)$

Hamiltonian Simulation Algorithms

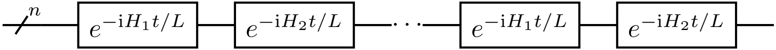
- Trotterization (= Product Formulae = Time/Operator Splitting)
  1st-order Trotter formula (Lie-Trotter) for $H = H_1 + H_2$

$$e^{-\mathrm{i}Ht} \approx \left(e^{-\mathrm{i}H_2 t/L} e^{-\mathrm{i}H_1 t/L}\right)^L$$

Cost/Complexity? error estimate and circuit implementation

$$e^{-\mathrm{i}Ht} = \left(e^{-\mathrm{i}H_2 t/L} e^{-\mathrm{i}H_1 t/L}\right)^L + \mathcal{O}\left(\|[H_1, H_2]\| t^2/L\right)$$

The number of Trotter steps $L = \mathcal{O}\left(\|[H_1, H_2]\| t^2/\epsilon\right)$



Hamiltonian Simulation Algorithms

- Trotterization (= Product Formulae = Time/Operator Splitting)
  1st-order Trotter formula (Lie-Trotter) for $H = H_1 + H_2$

  $$e^{-\mathrm{i}Ht} \approx \left(e^{-\mathrm{i}H_2 t/L} e^{-\mathrm{i}H_1 t/L}\right)^L$$

  Cost/Complexity? error estimate and circuit implementation

  $$e^{-\mathrm{i}Ht} = \left(e^{-\mathrm{i}H_2 t/L} e^{-\mathrm{i}H_1 t/L}\right)^L + \mathcal{O}\left(\|[H_1, H_2]\| t^2/L\right)$$

  The number of Trotter steps $L = \mathcal{O}\left(\|[H_1, H_2]\| t^2/\epsilon\right)$

  $$\overset{n}{/} \text{---} \boxed{e^{-\mathrm{i}H_1 t/L}} \text{---} \boxed{e^{-\mathrm{i}H_2 t/L}} \text{---} \cdots \text{---} \boxed{e^{-\mathrm{i}H_1 t/L}} \text{---} \boxed{e^{-\mathrm{i}H_2 t/L}} \text{---}$$

  $\Rightarrow \mathcal{O}\left(\|[H_1, H_2]\| t^2/\epsilon\right)$ queries to $e^{-\mathrm{i}H_1 s}$ and $e^{-\mathrm{i}H_2 s}$.

Hamiltonian Simulation Algorithms

- Trotterization (= Product Formulae = Time/Operator Splitting)
  1st-order Trotter formula (Lie-Trotter) for $H = H_1 + H_2$

$$e^{-\mathrm{i}Ht} \approx \left(e^{-\mathrm{i}H_2 t/L} e^{-\mathrm{i}H_1 t/L}\right)^L$$

  Cost/Complexity? error estimate and circuit implementation

$$e^{-\mathrm{i}Ht} = \left(e^{-\mathrm{i}H_2 t/L} e^{-\mathrm{i}H_1 t/L}\right)^L + \mathcal{O}\left(\|[H_1, H_2]\| t^2/L\right)$$

  The number of Trotter steps $L = \mathcal{O}\left(\|[H_1, H_2]\| t^2/\epsilon\right)$

$$\overset{n}{/}\!-\!\boxed{e^{-\mathrm{i}H_1 t/L}}-\boxed{e^{-\mathrm{i}H_2 t/L}}-\cdots-\boxed{e^{-\mathrm{i}H_1 t/L}}-\boxed{e^{-\mathrm{i}H_2 t/L}}-$$

  $\Rightarrow \mathcal{O}\left(\|[H_1, H_2]\| t^2/\epsilon\right)$ queries to $e^{-\mathrm{i}H_1 s}$ and $e^{-\mathrm{i}H_2 s}$.

  High order (p-th): query complexity $\mathcal{O}\left(\alpha_H t^{1+1/p}/\epsilon^{1/p}\right)$.

Hamiltonian Simulation Algorithms

- Trotterization (= Product Formulae = Time/Operator Splitting)
  1st-order Trotter formula (Lie-Trotter) for $H = H_1 + H_2$

  $$\overset{n}{/}\!-\boxed{e^{-\mathrm{i}H_1 t/L}}-\boxed{e^{-\mathrm{i}H_2 t/L}}-\cdots-\boxed{e^{-\mathrm{i}H_1 t/L}}-\boxed{e^{-\mathrm{i}H_2 t/L}}-$$

  High order (p-th): query complexity $\mathcal{O}\left(\alpha_H t^{1+1/p}/\epsilon^{1/p}\right)$.
  - Everything is unitary! No ancilla needed.
  - But it needs $e^{-\mathrm{i}H_j s}$ efficiently implementable.

Hamiltonian Simulation Algorithms

- Trotterization (= Product Formulae = Time/Operator Splitting)
  1st-order Trotter formula (Lie-Trotter) for $H = H_1 + H_2$

$$\overset{n}{\not\text{—}}\ \boxed{e^{-\mathrm{i}H_1 t/L}} - \boxed{e^{-\mathrm{i}H_2 t/L}} - \cdots - \boxed{e^{-\mathrm{i}H_1 t/L}} - \boxed{e^{-\mathrm{i}H_2 t/L}} -$$

  High order (p-th): query complexity $\mathcal{O}\left(\alpha_H t^{1+1/p}/\epsilon^{1/p}\right)$.
  - Everything is unitary! No ancilla needed.
  - But it needs $e^{-\mathrm{i}H_j s}$ efficiently implementable.

- Post-Trotter, e.g., truncated Taylor series, quantum signal processing (QSP), quantum singular value transformation (QSVT), etc.

$$e^{-\mathrm{i}Ht} \approx \sum_{k=0}^{K} \frac{(-\mathrm{i}Ht)^k}{k!} = \sum_{k=0}^{K} \sum_{\ell_1,\cdots,\ell_k} \frac{(-\mathrm{i}t)^k}{k!} H_{\ell_1} H_{\ell_2} \cdots H_{\ell_k}.$$

Upshot: $\Rightarrow \mathcal{O}(t \log(1/\epsilon))$

Hamiltonian Simulation Algorithms

- Trotterization (= Product Formulae = Time/Operator Splitting)
  1st-order Trotter formula (Lie-Trotter) for $H = H_1 + H_2$

$$\boxed{e^{-\mathrm{i}H_1 t/L}} - \boxed{e^{-\mathrm{i}H_2 t/L}} - \cdots - \boxed{e^{-\mathrm{i}H_1 t/L}} - \boxed{e^{-\mathrm{i}H_2 t/L}}$$

  High order (p-th): query complexity $\mathcal{O}\left(\alpha_H t^{1+1/p}/\epsilon^{1/p}\right)$.
  - Everything is unitary! No ancilla needed.
  - But it needs $e^{-\mathrm{i}H_j s}$ efficiently implementable.

- Post-Trotter, e.g., truncated Taylor series, quantum signal processing (QSP), quantum singular value transformation (QSVT), etc.

$$e^{-\mathrm{i}Ht} \approx \sum_{k=0}^{K} \frac{(-\mathrm{i}Ht)^k}{k!} = \sum_{k=0}^{K} \sum_{\ell_1,\cdots,\ell_k} \frac{(-\mathrm{i}t)^k}{k!} H_{\ell_1} H_{\ell_2} \cdots H_{\ell_k}.$$

Upshot: $\Rightarrow \mathcal{O}(t \log(1/\epsilon))$ $\Rightarrow$ Even better, say, $\mathcal{O}(t + \log(1/\epsilon))$?

Block-Encoding – Definition

Let A be a general $2^n \times 2^n$ matrix.

Idea:

$$U_A = \begin{pmatrix} A & * \\ * & * \end{pmatrix} \quad \rightarrow \text{ancilla qubits}$$

Block-Encoding – Definition

Let A be a general $2^n \times 2^n$ matrix. $\|A\| \leq \alpha$

Idea:

$$U_A = \begin{pmatrix} \frac{A}{\alpha} & * \\ * & * \end{pmatrix} \quad \to \text{ancilla qubits},$$

Block-Encoding – Definition

Let A be a general $2^n \times 2^n$ matrix. $\|A\| \le \alpha$

Idea:

$$U_A = \begin{pmatrix} \frac{\tilde{A}}{\alpha} & * \\ * & * \end{pmatrix} \qquad \begin{array}{l} \left\|\tilde{A} - A\right\| \le \epsilon \\ \to m \text{ ancilla qubits,} \end{array}$$

Block-Encoding – Definition

Let A be a general $2^n \times 2^n$ matrix. $\|A\| \leq \alpha$

Idea:

$$U_A = \begin{pmatrix} \frac{\tilde{A}}{\alpha} & * \\ * & * \end{pmatrix} \qquad \begin{matrix} \left\|\tilde{A} - A\right\| \leq \epsilon \\ \rightarrow m \text{ ancilla qubits,} \end{matrix}$$

Definition (Block-encoding)

U_A is an (α, m, ϵ)-block-encoding of A, if

$$\|A - \alpha\left(\langle 0^m| \otimes I_n\right) U_A \left(|0^m\rangle \otimes I_n\right)\| \leq \epsilon,$$

for some $\alpha \geq \|A\|$, $m > 0$ and $\epsilon > 0$. Here α is called the *subnormalization* factor and m is the number of ancilla qubits, and n is the number of system qubits. When $\epsilon = 0$, it is also called an (α, m)-block-encoding.

Block-Encoding – Definition

Let A be a general $2^n \times 2^n$ matrix. $\|A\| \leq \alpha$

Idea:

$$U_A = \begin{pmatrix} \frac{\tilde{A}}{\alpha} & * \\ * & * \end{pmatrix} \qquad \begin{array}{c} \left\|\tilde{A} - A\right\| \leq \epsilon \\ \rightarrow m \text{ ancilla qubits,} \end{array}$$

Definition (Block-encoding)

U_A is an (α, m, ϵ)-block-encoding of A, if

$$\|A - \alpha \left(\langle 0^m| \otimes I_n\right) U_A \left(|0^m\rangle \otimes I_n\right)\| \leq \epsilon,$$

for some $\alpha \geq \|A\|$, $m > 0$ and $\epsilon > 0$. Here α is called the *subnormalization* factor and m is the number of ancilla qubits, and n is the number of system qubits. When $\epsilon = 0$, it is also called an (α, m)-block-encoding.

Understanding: $U_A : 2^{m+n} \times 2^{m+n}$.

Block-Encoding – Definition

Let A be a general $2^n \times 2^n$ matrix. $\|A\| \leq \alpha$

Idea:

$$U_A = \begin{pmatrix} \frac{\tilde{A}}{\alpha} & * \\ * & * \end{pmatrix} \qquad \left\|\tilde{A} - A\right\| \leq \epsilon$$

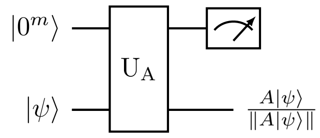
$\rightarrow m$ ancilla qubits,

Definition (Block-encoding)

U_A is an (α, m, ϵ)-block-encoding of A, if

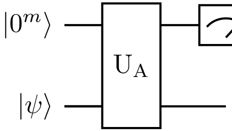
$$\|A - \alpha\left(\langle 0^m| \otimes I_n\right) U_A \left(|0^m\rangle \otimes I_n\right)\| \leq \epsilon,$$

for some $\alpha \geq \|A\|$, $m > 0$ and $\epsilon > 0$.



(upon getting 0 in measurement)

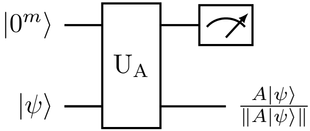
Block-Encoding – Definition cont'd



$\frac{A|\psi\rangle}{\|A|\psi\rangle\|}$ (upon getting 0 in measurement)

$$|0,\psi\rangle = |0\rangle \otimes |\psi\rangle = \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix}, \quad U_A\,|0,\psi\rangle = \begin{pmatrix} \frac{\tilde{A}}{\alpha} & * \\ * & * \end{pmatrix} \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{A}\,|\psi\rangle \\ * \end{pmatrix}.$$

Block-Encoding – Definition cont'd

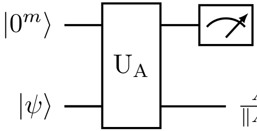


(upon getting 0 in measurement)

$$|0,\psi\rangle = |0\rangle \otimes |\psi\rangle = \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix}, \quad U_A\,|0,\psi\rangle = \begin{pmatrix} \frac{\tilde{A}}{\alpha} & * \\ * & * \end{pmatrix} \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{A}\,|\psi\rangle \\ * \end{pmatrix}.$$

Question: Well-defined? Not an empty set?

Block-Encoding – Definition cont'd



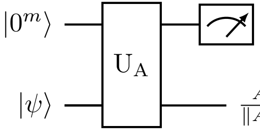
$\frac{A|\psi\rangle}{\|A|\psi\rangle\|}$ (upon getting 0 in measurement)

$$|0,\psi\rangle = |0\rangle \otimes |\psi\rangle = \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix}, \quad U_A\,|0,\psi\rangle = \begin{pmatrix} \frac{\tilde{A}}{\alpha} & * \\ * & * \end{pmatrix} \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{A}\,|\psi\rangle \\ * \end{pmatrix}.$$

Question: Well-defined? Not an empty set?

- Trivial example (unitary): U is a $(1, 0, 0)$-block-encoding of U.

Block-Encoding – Definition cont'd



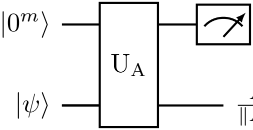
$\frac{A|\psi\rangle}{\|A|\psi\rangle\|}$ (upon getting 0 in measurement)

$$|0,\psi\rangle = |0\rangle \otimes |\psi\rangle = \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix}, \quad U_A\,|0,\psi\rangle = \begin{pmatrix} \frac{\tilde{A}}{\alpha} & * \\ * & * \end{pmatrix} \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{A}\,|\psi\rangle \\ * \end{pmatrix}.$$

Question: Well-defined? Not an empty set?

- Trivial example (unitary): U is a $(1,0,0)$-block-encoding of U.
- $(\alpha, 1)$-block-encoding is general. WLOG, assume $\|A\| \le 1$.

Block-Encoding – Definition cont'd



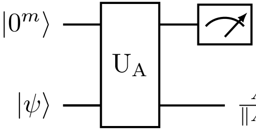
$\frac{A|\psi\rangle}{\|A|\psi\rangle\|}$ (upon getting 0 in measurement)

Question: Well-defined? Not an empty set?

- Trivial example (unitary): U is a $(1, 0, 0)$-block-encoding of U.
- $(\alpha, 1)$-block-encoding is general. WLOG, assume $\|A\| \leq 1$.
  Proof: $A = W\Sigma V^\dagger$. All singular values $\in [0, 1]$.

Reference: [Gilyen-Su-Low-Wiebe 2018/2019], Lecture notes by Lin Lin

Block-Encoding – Definition cont'd



$\frac{A|\psi\rangle}{\|A|\psi\rangle\|}$ (upon getting 0 in measurement)

Question: Well-defined? Not an empty set?

- Trivial example (unitary): U is a $(1,0,0)$-block-encoding of U.
- $(\alpha, 1)$-block-encoding is general. WLOG, assume $\|A\| \leq 1$.
  Proof: $A = W\Sigma V^\dagger$. All singular values $\in [0,1]$.

$$U_A := \begin{pmatrix} W & 0 \\ 0 & I_n \end{pmatrix} \begin{pmatrix} \Sigma & \sqrt{I_n - \Sigma^2} \\ \sqrt{I_n - \Sigma^2} & -\Sigma \end{pmatrix} \begin{pmatrix} V^\dagger & 0 \\ 0 & I_n \end{pmatrix}$$
$$= \begin{pmatrix} A & W\sqrt{I_n - \Sigma^2} \\ \sqrt{I_n - \Sigma^2}V^\dagger & -\Sigma \end{pmatrix}$$

Reference: [Gilyen-Su-Low-Wiebe 2018/2019], Lecture notes by Lin Lin

Block-Encoding – Properties

Properties: Let U_A be an (α, a, ϵ)-BE of A; U_B be a (β, b, δ)-BE of B

Block-Encoding – Properties

Properties: Let U_A be an (α, a, ϵ)-BE of A; U_B be a (β, b, δ)-BE of B

1. (BE of cA)

Block-Encoding – Properties

Properties: Let U_A be an (α, a, ϵ)-BE of A; U_B be a (β, b, δ)-BE of B

1. (BE of cA) U_A is an $(c\alpha, a, c\epsilon)$-BE of cA.

Block-Encoding – Properties

Properties: Let U_A be an (α, a, ϵ)-BE of A; U_B be a (β, b, δ)-BE of B

1. (BE of cA) U_A is an $(c\alpha, a, c\epsilon)$-BE of cA.
2. (BE of AB)
   $W = (I_b \otimes U_A)(I_a \otimes U_B)$ is an $(\alpha\beta, a + b, \alpha\delta + \beta\epsilon)$-BE of AB.

Block-Encoding – Properties

Properties: Let U_A be an (α, a, ϵ)-BE of A; U_B be a (β, b, δ)-BE of B

1. (BE of cA) U_A is an $(c\alpha, a, c\epsilon)$-BE of cA.
2. (BE of AB)
   $W = (I_b \otimes U_A)(I_a \otimes U_B)$ is an $(\alpha\beta, a + b, \alpha\delta + \beta\epsilon)$-BE of AB.
   Proof:

$$\left\| AB - \alpha\beta(\langle 0|^{\otimes a+b} \otimes I)(I_b \otimes U_A)(I_a \otimes U_B)(|0\rangle^{\otimes a+b} \otimes I) \right\|$$
$$= \left\| AB - \underbrace{\alpha(\langle 0|^{\otimes a} \otimes I)U_A(|0\rangle^{\otimes a} \otimes I)}_{\tilde{A}}\underbrace{\beta(\langle 0|^{\otimes b} \otimes I)U_B(|0\rangle^{\otimes b} \otimes I)}_{\tilde{B}} \right\|$$

Block-Encoding – Properties

Properties: Let U_A be an (α, a, ϵ)-BE of A; U_B be a (β, b, δ)-BE of B

1. (BE of cA) U_A is an $(c\alpha, a, c\epsilon)$-BE of cA.
2. (BE of AB)
   $W = (I_b \otimes U_A)(I_a \otimes U_B)$ is an $(\alpha\beta, a+b, \alpha\delta + \beta\epsilon)$-BE of AB.
   Proof:

$$\left\| AB - \alpha\beta(\langle 0|^{\otimes a+b} \otimes I)(I_b \otimes U_A)(I_a \otimes U_B)(|0\rangle^{\otimes a+b} \otimes I) \right\|$$

$$= \left\| AB - \underbrace{\alpha(\langle 0|^{\otimes a} \otimes I)U_A(|0\rangle^{\otimes a} \otimes I)}_{\tilde{A}}\underbrace{\beta(\langle 0|^{\otimes b} \otimes I)U_B(|0\rangle^{\otimes b} \otimes I)}_{\tilde{B}} \right\|$$

$$= \left\| AB - \tilde{A}B + \tilde{A}B - \tilde{A}\tilde{B} \right\| = \left\| (A - \tilde{A})B + \tilde{A}(B - \tilde{B}) \right\|$$

Block-Encoding – Properties

Properties: Let U_A be an (α, a, ϵ)-BE of A; U_B be a (β, b, δ)-BE of B

1. (BE of cA) U_A is an $(c\alpha, a, c\epsilon)$-BE of cA.
2. (BE of AB)
   $W = (I_b \otimes U_A)(I_a \otimes U_B)$ is an $(\alpha\beta, a+b, \alpha\delta + \beta\epsilon)$-BE of AB.
   Proof:

$$
\begin{aligned}
&\left\| AB - \alpha\beta(\langle 0|^{\otimes a+b} \otimes I)(I_b \otimes U_A)(I_a \otimes U_B)(|0\rangle^{\otimes a+b} \otimes I) \right\| \\
=&\left\| AB - \underbrace{\alpha(\langle 0|^{\otimes a} \otimes I)U_A(|0\rangle^{\otimes a} \otimes I)}_{\tilde{A}} \underbrace{\beta(\langle 0|^{\otimes b} \otimes I)U_B(|0\rangle^{\otimes b} \otimes I)}_{\tilde{B}} \right\| \\
=&\left\| AB - \tilde{A}B + \tilde{A}B - \tilde{A}\tilde{B} \right\| = \left\| (A - \tilde{A})B + \tilde{A}(B - \tilde{B}) \right\| \\
\leq&\alpha\delta + \beta\epsilon.
\end{aligned}
$$

Block-Encoding – Properties

Properties: Let U_A be an (α, a, ϵ)-BE of A; U_B be a (β, b, δ)-BE of B

1. (BE of cA) U_A is an $(c\alpha, a, c\epsilon)$-BE of cA.
2. (BE of AB)
   $W = (I_b \otimes U_A)(I_a \otimes U_B)$ is an $(\alpha\beta, a + b, \alpha\delta + \beta\epsilon)$-BE of AB.
3. (BE of $A + B$)

Block-Encoding – Properties

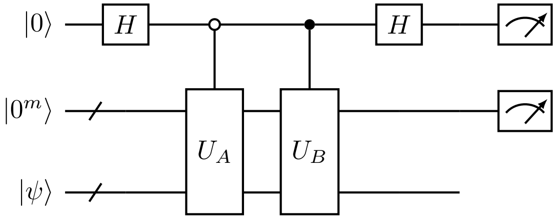
Properties: Let U_A be an (α, a, ϵ)-BE of A; U_B be a (β, b, δ)-BE of B

1. (BE of cA) U_A is an $(c\alpha, a, c\epsilon)$-BE of cA.
2. (BE of AB)
   $W = (I_b \otimes U_A)(I_a \otimes U_B)$ is an $(\alpha\beta, a + b, \alpha\delta + \beta\epsilon)$-BE of AB.
3. (BE of $A + B$) U_A: $(1, m, \epsilon)$-BE of A; U_B: $(1, m, \delta)$-BE of B

Block-Encoding – Properties

Properties: Let U_A be an (α, a, ϵ)-BE of A; U_B be a (β, b, δ)-BE of B

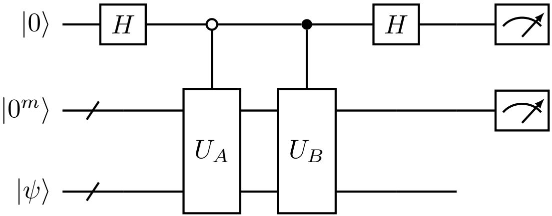
1. (BE of cA) U_A is an $(c\alpha, a, c\epsilon)$-BE of cA.
2. (BE of AB)
   $W = (I_b \otimes U_A)(I_a \otimes U_B)$ is an $(\alpha\beta, a + b, \alpha\delta + \beta\epsilon)$-BE of AB.
3. (BE of $A + B$) U_A: $(1, m, \epsilon)$-BE of A; U_B: $(1, m, \delta)$-BE of B
   The following circuit constructs a $(2, m, \delta + \epsilon)$-BE of $A + B$.



Block-Encoding – Properties

Properties: Let U_A be an (α, a, ϵ)-BE of A; U_B be a (β, b, δ)-BE of B

1. (BE of cA) U_A is an $(c\alpha, a, c\epsilon)$-BE of cA.
2. (BE of AB)
   $W = (I_b \otimes U_A)(I_a \otimes U_B)$ is an $(\alpha\beta, a+b, \alpha\delta + \beta\epsilon)$-BE of AB.
3. (BE of $A+B$) U_A: $(1, m, \epsilon)$-BE of A; U_B: $(1, m, \delta)$-BE of B
   The following circuit constructs a $(2, m, \delta + \epsilon)$-BE of $A + B$.



More generally, linear combination of block-encodings can be constructed via Linear Combination of Unitaries (LCU) Lemma.

LCU Lemma

LCU Lemma: $T = \sum_{j \in [L]} c_j U_j$ for unitaries U_j. $\|c\|_1 = \sum_{j \in [L]} |c_j|$.

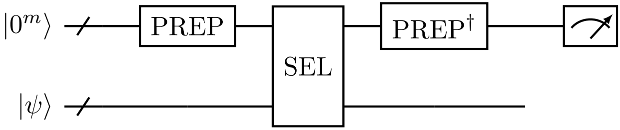
LCU [Berry-Childs-Kothari 2015], General LCBE [Gilyen-Su-Low-Wiebe 2018]

LCU Lemma

LCU Lemma: $T = \sum_{j\in[L]} c_j U_j$ for unitaries U_j. $\|c\|_1 = \sum_{j\in[L]} |c_j|$.

One can get a $(\|c\|_1, \lceil \log_2 L \rceil)$-block-encoding by:

- $\mathrm{SEL} := \sum_{j\in[L]} |j\rangle\langle j| \otimes U_j$
- $\mathrm{PREP}\,|0\rangle = \frac{1}{\sqrt{\|c\|_1}} \sum_{j\in[L]} \sqrt{c_j}\,|j\rangle$.



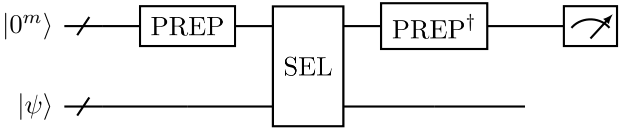
LCU [Berry-Childs-Kothari 2015], General LCBE [Gilyen-Su-Low-Wiebe 2018]

LCU Lemma

LCU Lemma: $T = \sum_{j \in [L]} c_j U_j$ for unitaries U_j. $\|c\|_1 = \sum_{j\in[L]} |c_j|$.

One can get a $(\|c\|_1, \lceil \log_2 L \rceil)$-block-encoding by:

- $\mathrm{SEL} := \sum_{j \in [L]} |j\rangle \langle j| \otimes U_j$
- $\mathrm{PREP}\, |0\rangle = \frac{1}{\sqrt{\|c\|_1}} \sum_{j \in [L]} \sqrt{c_j}\, |j\rangle$.



General LCBE: $\max_j m_j + \lceil \log_2 L \rceil$ ancillas

LCU [Berry-Childs-Kothari 2015], General LCBE [Gilyen-Su-Low-Wiebe 2018]

Matrix Function

We can "$+$" and "$\times$" $\Rightarrow$ we can BE $\mathrm{poly}(A)$

Matrix Function

We can "$+$" and "$\times$" $\Rightarrow$ we can BE $\mathrm{poly}(A)$

$\Rightarrow$ $\boxed{\text{We can BE } f(A).}$ Super Powerful!!!

Matrix Function

We can "$+$" and "$\times$" $\Rightarrow$ we can BE $\mathrm{poly}(A)$

$\Rightarrow$ $\boxed{\text{We can BE } f(A).}$ Super Powerful!!!

e.g., e^{-iHt} Hamiltonian Simulation, $e^{-\beta H}$ Gibbs distribution, A^{-1} matrix inversion, etc.

Matrix Function

We can "+" and "$\times$" $\Rightarrow$ we can BE $\mathrm{poly}(A)$

$\Rightarrow$ $\boxed{\text{We can BE } f(A).}$ Super Powerful!!!

e.g., e^{-iHt} Hamiltonian Simulation, $e^{-\beta H}$ Gibbs distribution, A^{-1} matrix inversion, etc.

But $A + A^2 + \cdots + A^d$
Number of ancillas: $m + 2m + \cdots + dm \Rightarrow dm + \log(d)$ HUGE!

Question: Can we do better?

Matrix Function

We can “$+$” and “$\times$” $\Rightarrow$ we can BE $\mathrm{poly}(A)$

$\Rightarrow$ $\boxed{\text{We can BE } f(A).}$ Super Powerful!!!

e.g., e^{-iHt} Hamiltonian Simulation, $e^{-\beta H}$ Gibbs distribution, A^{-1} matrix inversion, etc.

But $A + A^2 + \cdots + A^d$
Number of ancillas: $m + 2m + \cdots + dm \Rightarrow dm + \log(d)$ HUGE!

Question: Can we do better? Yes! 1 additional ancilla is sufficient!

Quantum Singular Value Transformation (QSVT) / Quantum Signal Processing (QSP)

QSVT

$A = W\Sigma V^\dagger$ $\quad f^\diamond(A) := W f(\Sigma) V^\dagger$ Generalized Matrix Function

QSVT

$A = W\Sigma V^\dagger \quad f^\diamond(A) := W f(\Sigma) V^\dagger$ Generalized Matrix Function

Theorem (QSVT with odd real polynomial)

Let U_A be a $(1, m)$-block-encoding of $A \in \mathbb{C}^{2^n \times 2^n}$. Given an odd polynomial $P_\Re(x) \in \mathbb{R}[x]$ of odd degree d satisfying

$$|P_\Re(x)| \leqslant 1, \forall x \in [-1, 1].$$

We can find a sequence of phase factors $\Phi \in \mathbb{R}^{d+1}$ and construct a $(1, m+1)$-block-encoding of $P_\Re^\diamond(A)$ that uses $U_A, U_A^\dagger$, m-qubit controlled NOT, and single qubit rotation gates for $\mathcal{O}(d)$ times.

QSVT

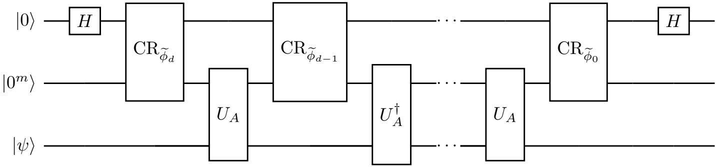
$A = W\Sigma V^\dagger \quad f^\diamond(A) := W f(\Sigma) V^\dagger$ Generalized Matrix Function

Theorem (QSVT with odd real polynomial)

Let U_A be a $(1, m)$-block-encoding of $A \in \mathbb{C}^{2^n \times 2^n}$. Given an odd polynomial $P_{\Re}(x) \in \mathbb{R}[x]$ of odd degree d satisfying

$$|P_{\Re}(x)| \leqslant 1, \forall x \in [-1, 1].$$

We can find a sequence of phase factors $\Phi \in \mathbb{R}^{d+1}$ and construct a $(1, m+1)$-block-encoding of $P_{\Re}^{\diamond}(A)$ that uses $U_A, U_A^\dagger$, m-qubit controlled NOT, and single qubit rotation gates for $\mathcal{O}(d)$ times.



Optimal Hamiltonian Simulation

Given U_H: an $(\alpha, m, 0)$-block-encoding of H.
Goal: an algorithm that makes $\mathcal{O}(t + \log(1/\epsilon))$ queries to U_H.

Optimal Hamiltonian Simulation

Given U_H: an $(\alpha, m, 0)$-block-encoding of H.
Goal: an algorithm that makes $\mathcal{O}(t + \log(1/\epsilon))$ queries to U_H.

- $e^{iHt} = e^{i\frac{H}{\alpha}\alpha t}$. WLOG, assume $\alpha = 1$.

Optimal Hamiltonian Simulation

Given U_H: an $(\alpha, m, 0)$-block-encoding of H.
Goal: an algorithm that makes $\mathcal{O}(t + \log(1/\epsilon))$ queries to U_H.

- $e^{iHt} = e^{i\frac{H}{\alpha}\alpha t}$. WLOG, assume $\alpha = 1$. $e^{itx} = \cos(tx) + i\sin(tx)$

Optimal Hamiltonian Simulation

Given U_H: an $(\alpha, m, 0)$-block-encoding of H.
Goal: an algorithm that makes $\mathcal{O}(t + \log(1/\epsilon))$ queries to U_H.

- $e^{iHt} = e^{i\frac{H}{\alpha}\alpha t}$. WLOG, assume $\alpha = 1$. $e^{itx} = \cos(tx) + i\sin(tx)$
- Jacobi-Anger expansion on $[-1, 1]$:

$$\cos(tx) = J_0(t) + 2\sum_{k=1}^{\infty}(-1)^k J_{2k}(t)T_{2k}(x),$$

$$\sin(tx) = 2\sum_{k=0}^{\infty}(-1)^k J_{2k+1}(t)T_{2k+1}(x).$$

$J_\nu(t)$ denotes Bessel functions of the first kind.

Optimal Hamiltonian Simulation

Given U_H: an $(\alpha, m, 0)$-block-encoding of H.
Goal: an algorithm that makes $\mathcal{O}(t + \log(1/\epsilon))$ queries to U_H.

- $e^{iHt} = e^{i\frac{H}{\alpha}\alpha t}$. WLOG, assume $\alpha = 1$. $e^{itx} = \cos(tx) + i\sin(tx)$
- Jacobi-Anger expansion on $[-1, 1]$:

$$\cos(tx) = J_0(t) + 2\sum_{k=1}^{\infty}(-1)^k J_{2k}(t)T_{2k}(x),$$

$$\sin(tx) = 2\sum_{k=0}^{\infty}(-1)^k J_{2k+1}(t)T_{2k+1}(x).$$

- This series converges rapidly. Truncating it with

$$r = \Theta\left(t + \frac{\log(1/\epsilon)}{\log(e + \log(1/\epsilon)/t)}\right)$$

terms gives a polynomial approximation (with precision ϵ and degree $2r+1$) of $\cos(tx) + i\sin(tx) = e^{itx}$.

Optimal Hamiltonian Simulation

Given U_H: an $(\alpha, m, 0)$-block-encoding of H.
Goal: an algorithm that makes $\mathcal{O}(t + \log(1/\epsilon))$ queries to U_H.

- $e^{iHt} = e^{i\frac{H}{\alpha}\alpha t}$. WLOG, assume $\alpha = 1$. $e^{itx} = \cos(tx) + i\sin(tx)$

Query Complexity: ($\alpha \geq \|H\|$.)

$$\mathcal{O}\left(\alpha t + \frac{\log(1/\epsilon)}{\log(e + \log(1/\epsilon)/t)}\right).$$

Success Probability

e^{-iHt} is unitary. Success probability upon measurement is ok. [1]

[1] In fact, it is an issue even for Hamiltonian simulation that leads to exponential cost in time. But for unitary dynamics, OAA (can be viewed as a form of QSVT) can solve the issue.

Success Probability

e^{-iHt} is unitary. Success probability upon measurement is ok. [1]

But BE of non-unitary. Success probability can be very small!!

[1] In fact, it is an issue even for Hamiltonian simulation that leads to exponential cost in time. But for unitary dynamics, OAA (can be viewed as a form of QSVT) can solve the issue.

Success Probability

e^{-iHt} is unitary. Success probability upon measurement is ok. [1]

But BE of non-unitary. Success probability can be very small!!

Suppose $|\psi\rangle$ can be prepared by U_ψ, i.e., $U_\psi |0^n\rangle = |\psi\rangle$, and

$$|\psi\rangle = \sqrt{p}\,|\psi_{\text{good}}\rangle + \sqrt{1-p}\,|\psi_{\text{bad}}\rangle\,.$$

The success probability of getting ψ_{good} is p.

$\Rightarrow$ need to repeat measurement $\mathcal{O}(1/p)$ times.

[1] In fact, it is an issue even for Hamiltonian simulation that leads to exponential cost in time. But for unitary dynamics, OAA (can be viewed as a form of QSVT) can solve the issue.

Success Probability

e^{-iHt} is unitary. Success probability upon measurement is ok. [1]

But BE of non-unitary. Success probability can be very small!!

Suppose $|\psi\rangle$ can be prepared by U_ψ, i.e., $U_\psi |0^n\rangle = |\psi\rangle$, and

$$|\psi\rangle = \sqrt{p}\,|\psi_{\text{good}}\rangle + \sqrt{1-p}\,|\psi_{\text{bad}}\rangle\,.$$

The success probability of getting ψ_{good} is p.

$\Rightarrow$ need to repeat measurement $\mathcal{O}(1/p)$ times.

What if p is too small?

[1] In fact, it is an issue even for Hamiltonian simulation that leads to exponential cost in time. But for unitary dynamics, OAA (can be viewed as a form of QSVT) can solve the issue.

Success Probability

e^{-iHt} is unitary. Success probability upon measurement is ok. [1]

But BE of non-unitary. Success probability can be very small!!

Suppose $|\psi\rangle$ can be prepared by U_ψ, i.e., $U_\psi |0^n\rangle = |\psi\rangle$, and

$$|\psi\rangle = \sqrt{p}\,|\psi_{\text{good}}\rangle + \sqrt{1-p}\,|\psi_{\text{bad}}\rangle .$$

The success probability of getting ψ_{good} is p.

$\Rightarrow$ need to repeat measurement $\mathcal{O}(1/p)$ times.

What if p is too small?

Amplitude Amplification (AA): The success probability can be boosted from p to $\Omega(1)$ via AA that accesses $\mathcal{O}(1/\sqrt{p})$ times of the circuit U.

[1] In fact, it is an issue even for Hamiltonian simulation that leads to exponential cost in time. But for unitary dynamics, OAA (can be viewed as a form of QSVT) can solve the issue.

Summary of Part 2

- Hamiltonian simulation and Trotterization
- Block-encoding: Definition and Properties
- LCU and QSVT
- Optimal Hamiltonian Simulation via QSVT
- Success Probability

Other advanced topics

- General Linear Differential Equation *(optional)*
- Hamiltonian Simulation – time dependent case? with unboundeded operator? *(workshop talk)*

References on Quantum Algorithms

- M. Nielsen and I. Chuang. Quantum Computation and Quantum Information, Cambridge University Press *(classics)*
- Lecture Notes on Quantum Computation by Umesh Vazirani (UC Berkeley) *(entry level)*
- Lecture Notes on Quantum Computation by John Preskill (Caltech) *(entry level)*
- Lecture Notes on Quantum Computation by Ryan O'Donnell (CMU) *(entry level)*
- Lecture notes on Quantum Algorithms for Scientific Computations by Lin Lin (UC Berkeley) [arXiv:2201.08309] *(advanced topics)*
- Lecture notes on Quantum Algorithms by Andrew Childs (U Maryland) *(advanced topics)*
- Qiskit Textbook by IBM (https://qiskit.org/learn) *(Algorithm Demos)*

Thank you for your attention!