Introduction to Quantum Computing

Rafael Nepomechie
Physics Department
University of Miami

Mathematics & Physics of Integrability

MATRIX, Creswick, Australia

July 2024

Outline

1. Basics
2. Some “textbook” quantum algorithms
   - Simon
   - Quantum Fourier Transform
   - Quantum Phase Estimation
3. Towards physics applications
   - [Quantum state preparation - next lecture!]
   - Variational Quantum Eigensolver
   - Quantum dynamics
4. Conclusions

1. Basics

"qubit" : 2-state system

basis: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

"Quantum computer":

- device with n qubits

"computational basis"

basis: $|x\rangle_n = |x_{n-1}\rangle \otimes \cdots \otimes |x_0\rangle$ $x_j \in \{0,1\}$ binary bits $x = \sum_{j=0}^{n-1} x_j 2^j$ integer $0 \leq x < 2^n$

"qubit" : 2-state system

basis: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

"Quantum computer":

- device with n qubits

"computational basis"

basis: $|x\rangle_n = |x_{n-1}\rangle \otimes \cdots \otimes |x_0\rangle$ $x_j \in \{0, 1\}$ $x = \sum_{j=0}^{n-1} x_j 2^j$ integer $0 \leq x < 2^n$

binary bits

Tensor products are often suppressed!

$$|x\rangle_n = |x_{n-1}\rangle \cdots |x_0\rangle$$

"qubit" : 2-state system

basis: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

"Quantum computer":

- device with n qubits

"computational basis"

basis: $|x\rangle_n = |x_{n-1}\rangle \otimes \cdots \otimes |x_0\rangle$ $x_j \in \{0,1\}$ $x = \sum_{j=0}^{n-1} x_j 2^j$ integer $0 \le x < 2^n$

binary bits

- can initialize each qubit $|0\rangle^{\otimes n}$

- can perform unitary transformations on qubits (decomposed into 1-qubit & 2-qubit unitary "gates")

Example: NOT $X = \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ $X|0\rangle = |1\rangle$ $X|1\rangle = |0\rangle$ $X|x\rangle = |x \oplus 1\rangle$ $x \in \{0,1\}$

$\oplus$: addition mod 2

"qubit" : 2-state system

basis: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

"Quantum computer":

- device with n qubits

"computational basis"

basis: $|x\rangle_n = |x_{n-1}\rangle \otimes \cdots \otimes |x_0\rangle$ $x_j \in \{0,1\}$ binary bits $x = \sum_{j=0}^{n-1} x_j 2^j$ integer $0 \le x < 2^n$

- can initialize each qubit $|0\rangle^{\otimes n}$

- can perform unitary transformations on qubits (decomposed into 1-qubit & 2-qubit unitary "gates")

Example: Hadamard $H = \frac{1}{\sqrt{2}}(X+Z) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

"qubit" : 2-state system

basis: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

"Quantum computer":

- device with n qubits

"computational basis"

basis: $|x\rangle_n = |x_{n-1}\rangle \otimes \cdots \otimes |x_0\rangle$ $x_j \in \{0,1\}$ binary bits $x = \sum_{j=0}^{n-1} x_j 2^j$ integer $0 \le x < 2^n$

- can initialize each qubit $|0\rangle^{\otimes n}$

- can perform unitary transformations on qubits (decomposed into 1-qubit & 2-qubit unitary "gates")

"target" changes to $X|y\rangle$ if $x = 1$

Example: CNOT $C^X|x\rangle|y\rangle = |x\rangle|x \oplus y\rangle$

"control" does not change

$x, y \in \{0,1\}$

$\oplus$: addition mod 2

"qubit" : 2-state system

basis: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

"Quantum computer":

- device with n qubits

"computational basis"

basis: $|x\rangle_n = |x_{n-1}\rangle \otimes \cdots \otimes |x_0\rangle$ $x_j \in \{0,1\}$ $x = \sum_{j=0}^{n-1} x_j 2^j$ integer $0 \le x < 2^n$

binary bits

- can initialize each qubit $|0\rangle^{\otimes n}$
- can perform unitary transformations on qubits (decomposed into 1-qubit & 2-qubit unitary "gates")
- can perform projective measurements of $\sum_{0 \le x < 2^n} x|x\rangle\langle x|$ $|\Psi\rangle_n = \sum_{0 \le x < 2^n} \alpha_x |x\rangle \mapsto |x\rangle$

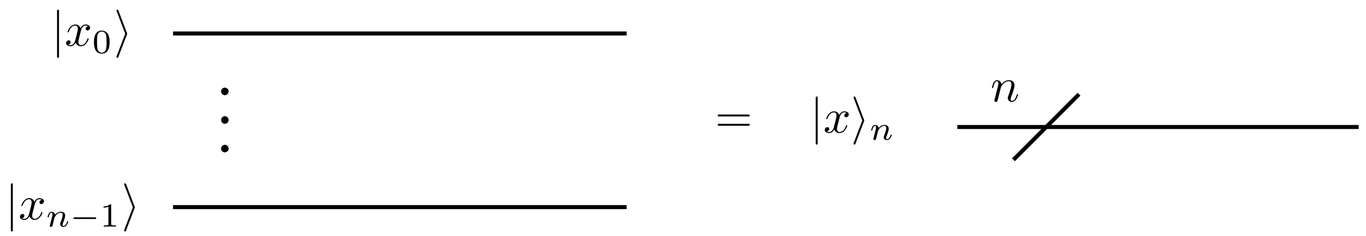
"quantum circuit"

probability $|\alpha_x|^2$

"Circuit diagrams":

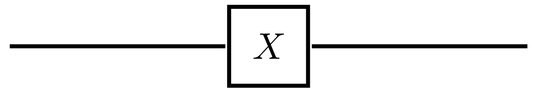
- represent qubits by horizontal "wires" :

$$|x\rangle_n = |x_{n-1}\rangle \otimes \cdots \otimes |x_0\rangle$$

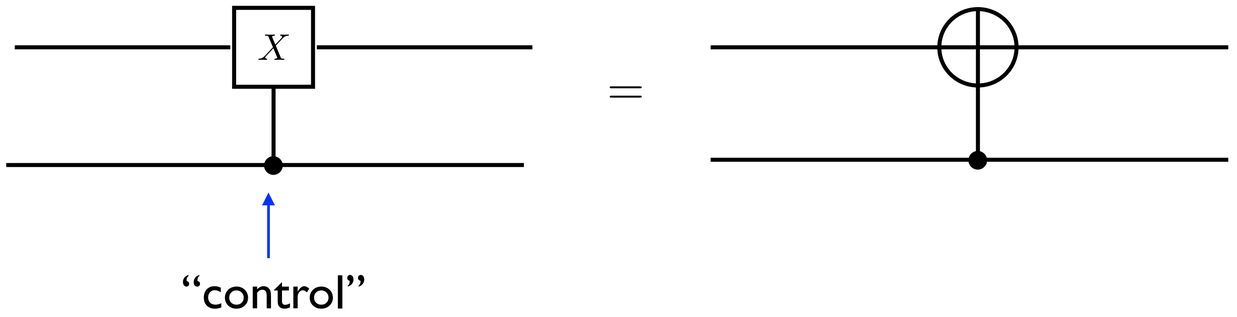


- represent gates symbolically:

Examples: X



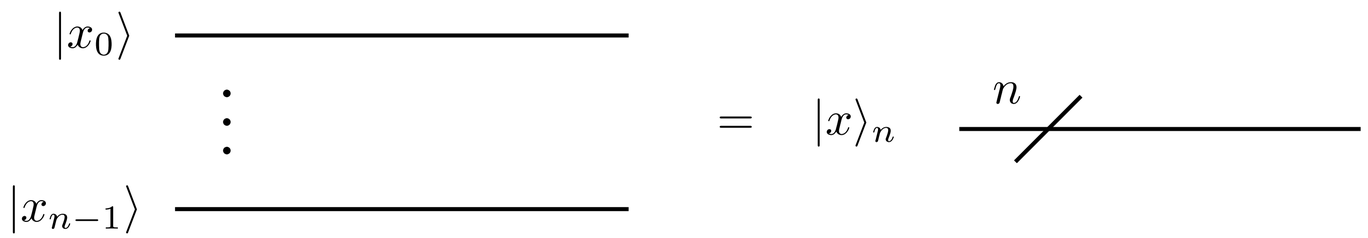
C^X



“Circuit diagrams”:

- represent qubits by horizontal “wires” :

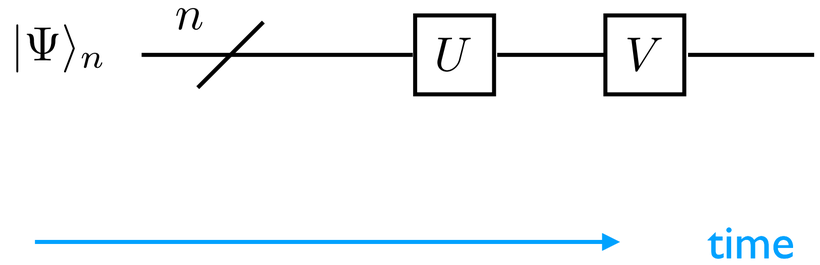
$$|x\rangle_n = |x_{n-1}\rangle \otimes \cdots \otimes |x_0\rangle$$



- represent gates symbolically:

- “time” flows to the right:

$$V\,U\,|\Psi\rangle_n$$



Currently: $n \sim 10^2$ IBM, Google, … “noisy” - make many errors! Noisy Intermediate-Scale Quantum era

Dream: $n \sim 10^4$ fault-tolerant

In the meantime, can test algorithms (“quantum circuits”) using noiseless *simulators*

e.g. IBM Qiskit simulators $n \sim 30$

2. Some “textbook” quantum algorithms

- Simon

x : n-bit integer

$f(x)$: n-bit integer $\quad f(x) = f(x \oplus a)$ for all x "periodic" $\quad \oplus$: bitwise addition mod 2

$$5 \oplus 3 : \quad 101_2 \oplus 011_2 = 110_2 = 6$$

oracle
("black box")

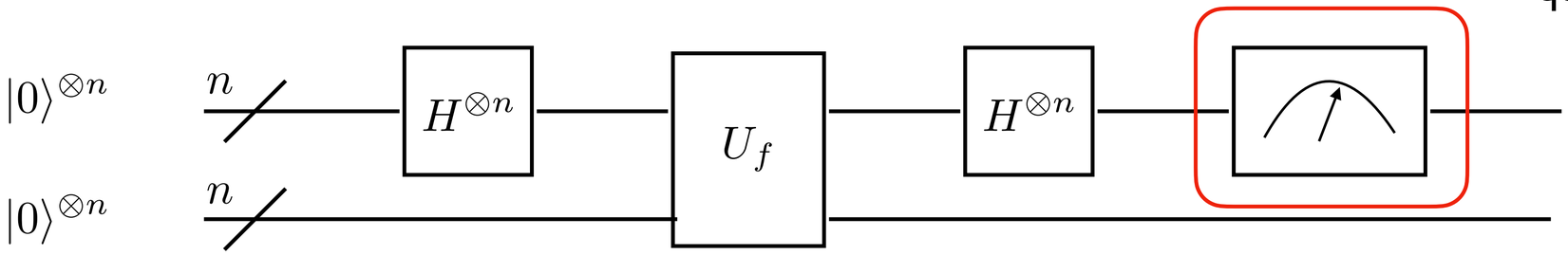
The problem: find the period a

Classical: call oracle ~ $2^{n/2}$ times

Quantum: call oracle ~ n times Exponential advantage!

Unitary U_f $\quad U_f\left(|y\rangle_n|x\rangle_n\right) = |y \oplus f(x)\rangle_n|x\rangle_n$ for all n-bit integers x, y

quantum implementation of oracle



$$\left(\mathbb{I}^{\otimes n} \otimes H^{\otimes n}\right) U_f \left(\mathbb{I}^{\otimes n} \otimes H^{\otimes n}\right) |0\rangle_n|0\rangle_n$$

$$= \cdots = \frac{1}{2^{n-1}} \sum_x \sum_{a \cdot y = 0} (-1)^{x \cdot y} |f(x)\rangle_n |y\rangle_n$$

$$x \cdot y \equiv \bigoplus_{j=0}^{n-1} x_j \, y_j = x_0 \, y_0 \oplus x_1 \, y_1 \oplus \cdots \oplus x_{n-1} \, y_{n-1} \in \{0, 1\}$$

Measure "top" n qubits

We learn a (random) value of y satisfying $\quad a \cdot y = 0$

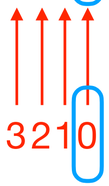
Can determine a with ~ n such y-values! (~ n equations for n unknowns $a_0, a_1, \ldots, a_{n-1}$)

"classical post-processing"

Implementing U_f

Example: $f(x) = \begin{cases} x \oplus a & \text{if } x_J = 0 \\ x & \text{if } x_J = 1 \end{cases}$ where J is least index such that $a_J = 1$ $f(x) = f(x \oplus a)$

Taking $n = 4\,, \quad a = 9 = 1001_2 \quad \Rightarrow \quad J = 0$



Implementing U_f

Example: $f(x) = \begin{cases} x \oplus a & \text{if } x_J = 0 \\ x & \text{if } x_J = 1 \end{cases}$ where J is least index such that $a_J = 1$ $f(x) = f(x \oplus a)$

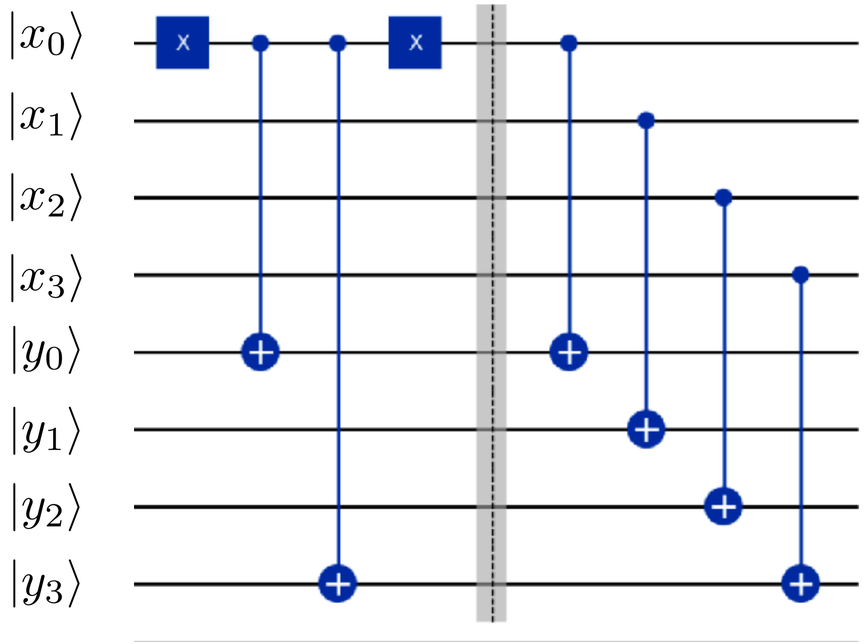
Taking $n = 4\,, \quad a = 9 = 1001_2 \quad \Rightarrow \quad J = 0 \quad \therefore \quad f(x) = \begin{cases} x \oplus 1001_2 & \text{if } x_0 = 0 \\ x & \text{if } x_0 = 1 \end{cases}$

Implementing U_f

Example: $f(x) = \begin{cases} x \oplus a & \text{if } x_J = 0 \\ x & \text{if } x_J = 1 \end{cases}$ where J is least index such that $a_J = 1$ $\qquad f(x) = f(x \oplus a)$

Taking $n = 4\,, \quad a = 9 = 1001_2 \quad \Rightarrow \quad J = 0 \quad \therefore \quad f(x) = \begin{cases} x \oplus 1001_2 & \text{if } x_0 = 0 \\ x & \text{if } x_0 = 1 \end{cases}$

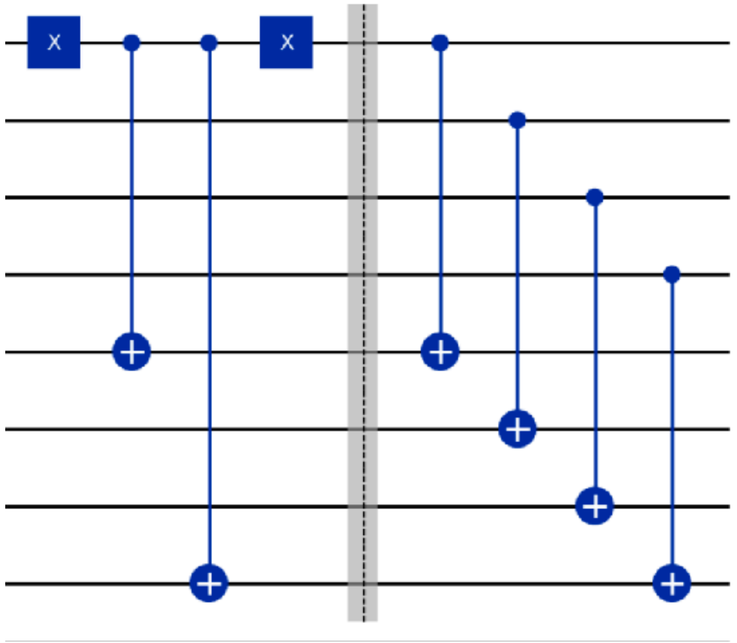
$U_f\left(|y\rangle_n |x\rangle_n\right) = |y \oplus f(x)\rangle_n |x\rangle_n$ for all n-bit integers x, y



$$\Big\} = \begin{cases} |y \oplus x \oplus 1001_2\rangle_4 & \text{if } x_0 = 0 \\ |y \oplus x\rangle_4 & \text{if } x_0 = 1 \end{cases}$$

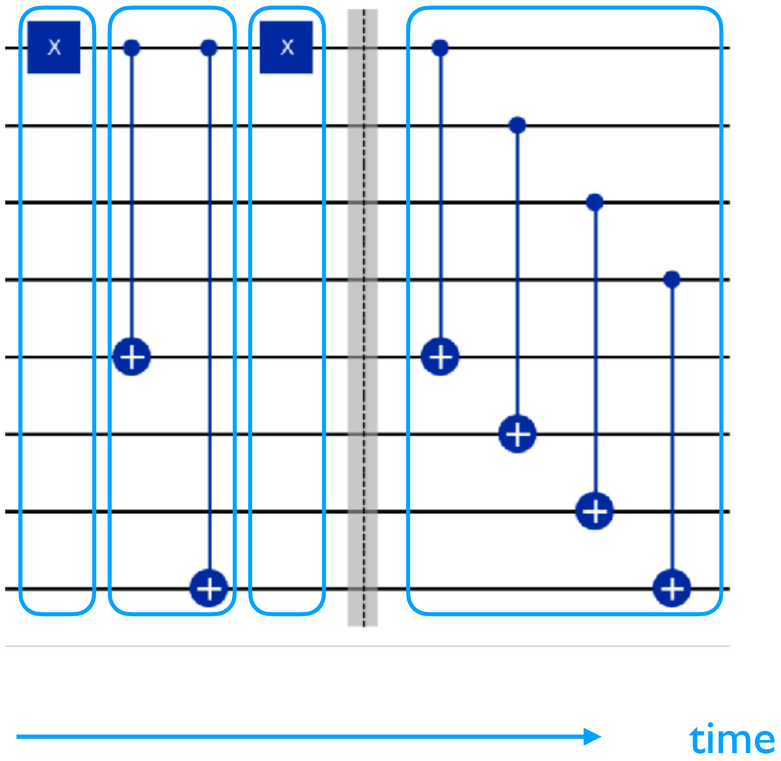
$$= |y \oplus f(x)\rangle_4 \qquad \checkmark$$

circuit *size* and *depth*



circuit *size:* number of elementary gates $= 8$

circuit *size* and *depth*



circuit *size:* number of elementary gates $= 8$

circuit depth: number of time steps $= 4$

In the NISQ era, most attention is on circuits whose size and depth does not grow too rapidly with the number of qubits

Remarks

- Simon's algorithm is contrived & useless, but it is a simple example of an algorithm with an *exponential* advantage

- This is a precursor to Shor's algorithm, which finds the period under ordinary addition

$\longrightarrow$ efficient factorization $N = pq$ $\longrightarrow$ breaking RSA encryption

- Quantum Fourier Transform

$$U_{FT}|x\rangle_n = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} e^{2\pi i x y/2^n} |y\rangle_n \qquad 0 \leq x < 2^n$$

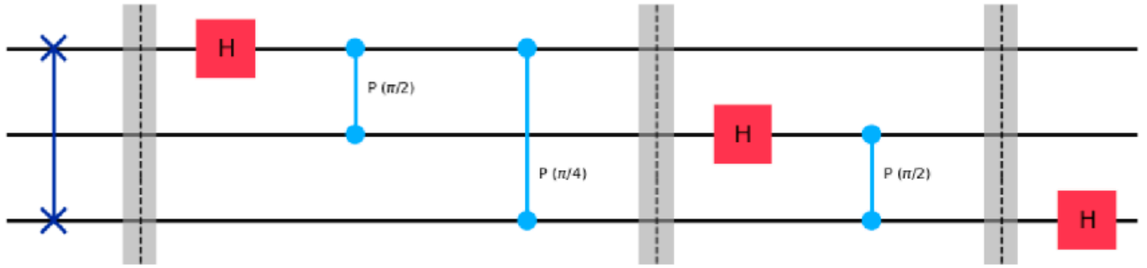
unitary

Examples:

$n = 1$:

$$U_{FT}|x\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + \underbrace{e^{\pi i x}}_{(-1)^x} |1\rangle\right) = H|x\rangle \qquad \Rightarrow \qquad U_{FT} = H$$

$n = 3$:



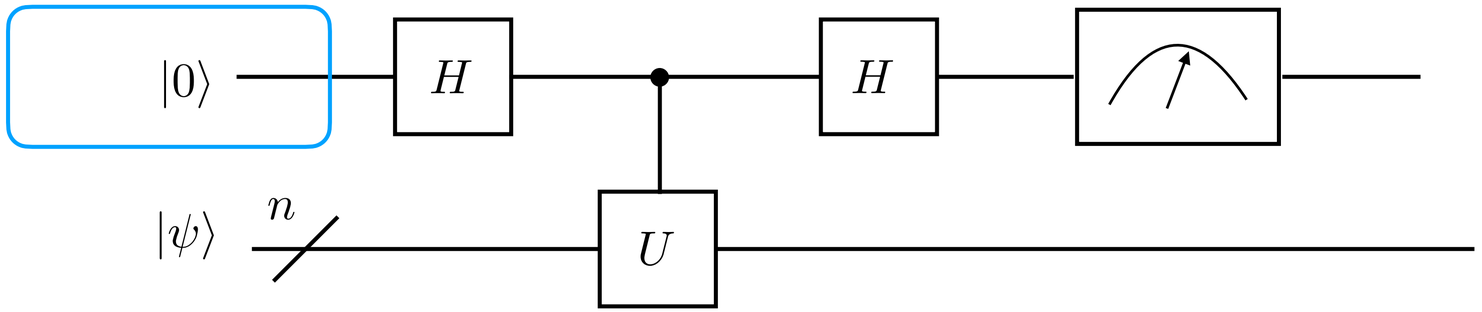
- Quantum Phase Estimation (QPE)

Unitary U , eigenvector $|\psi\rangle$

$$U|\psi\rangle = e^{i\theta}|\psi\rangle \qquad \theta \quad \text{real}$$

QPE: given U and $|\psi\rangle$, estimate θ

Baby version:



"Hadamard test"

"ancillary" (extra) qubit

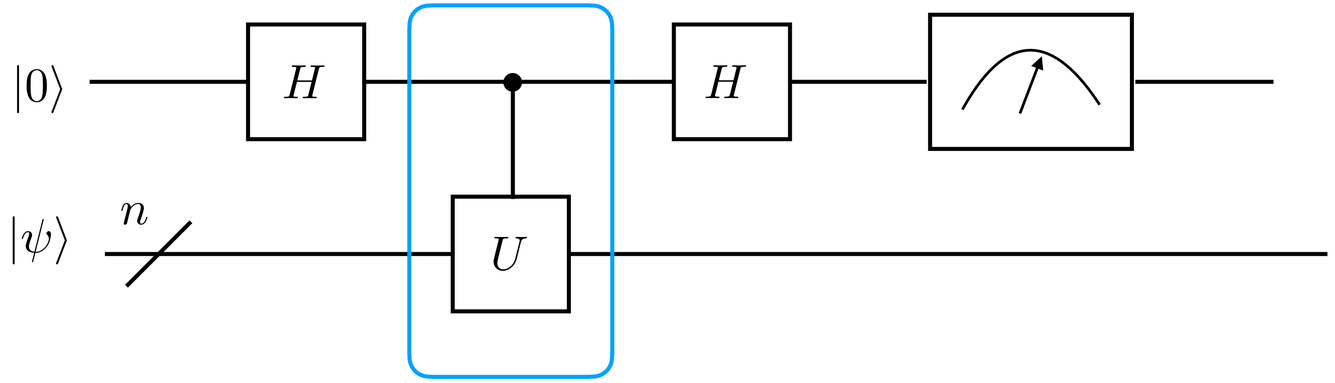
- Quantum Phase Estimation (QPE)

Unitary U , eigenvector $|\psi\rangle$

$$U|\psi\rangle = e^{i\theta}|\psi\rangle \qquad \theta \quad \text{real}$$

QPE: given U and $|\psi\rangle$, estimate θ

Baby version:



"Hadamard test"

Controlled version of U

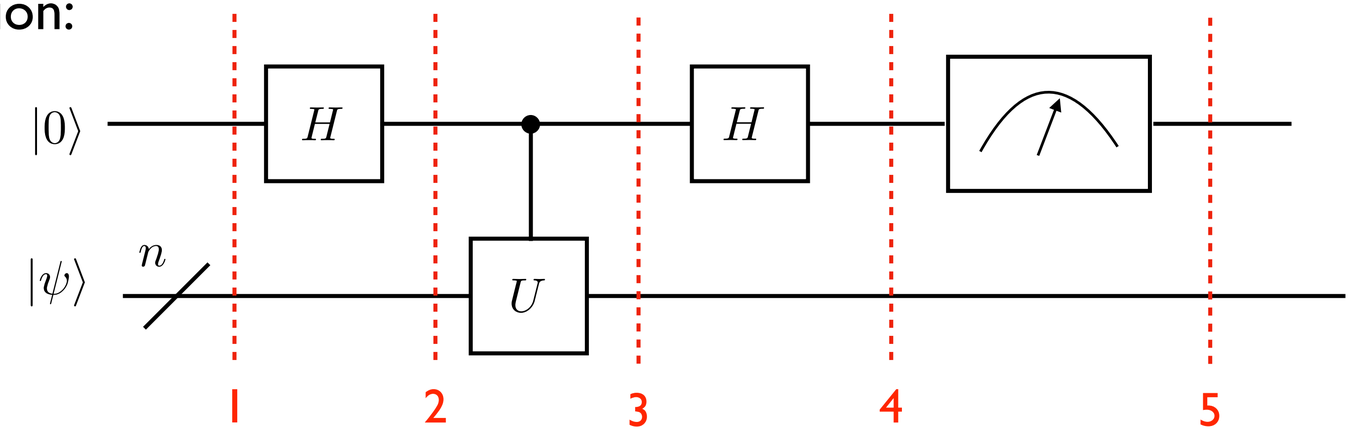
- Quantum Phase Estimation (QPE)

Unitary U , eigenvector $|\psi\rangle$

$$U|\psi\rangle = e^{i\theta}|\psi\rangle \qquad \theta \quad \text{real}$$

QPE: given U and $|\psi\rangle$, estimate θ

Baby version:



"Hadamard test"

1: $|\psi\rangle|0\rangle$

2: $|\psi\rangle\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle) = \frac{1}{\sqrt{2}}(|\psi\rangle|0\rangle+|\psi\rangle|1\rangle)$

3: $\frac{1}{\sqrt{2}}(|\psi\rangle|0\rangle+\underbrace{U|\psi\rangle}_{e^{i\theta}|\psi\rangle}|1\rangle)$

4: $\frac{1}{\sqrt{2}}\left[|\psi\rangle\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)+e^{i\theta}|\psi\rangle\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\right]$

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

5: $p(0) = |\frac{1}{2}\left(1+e^{i\theta}\right)|^2 = \cos^2(\theta/2)$

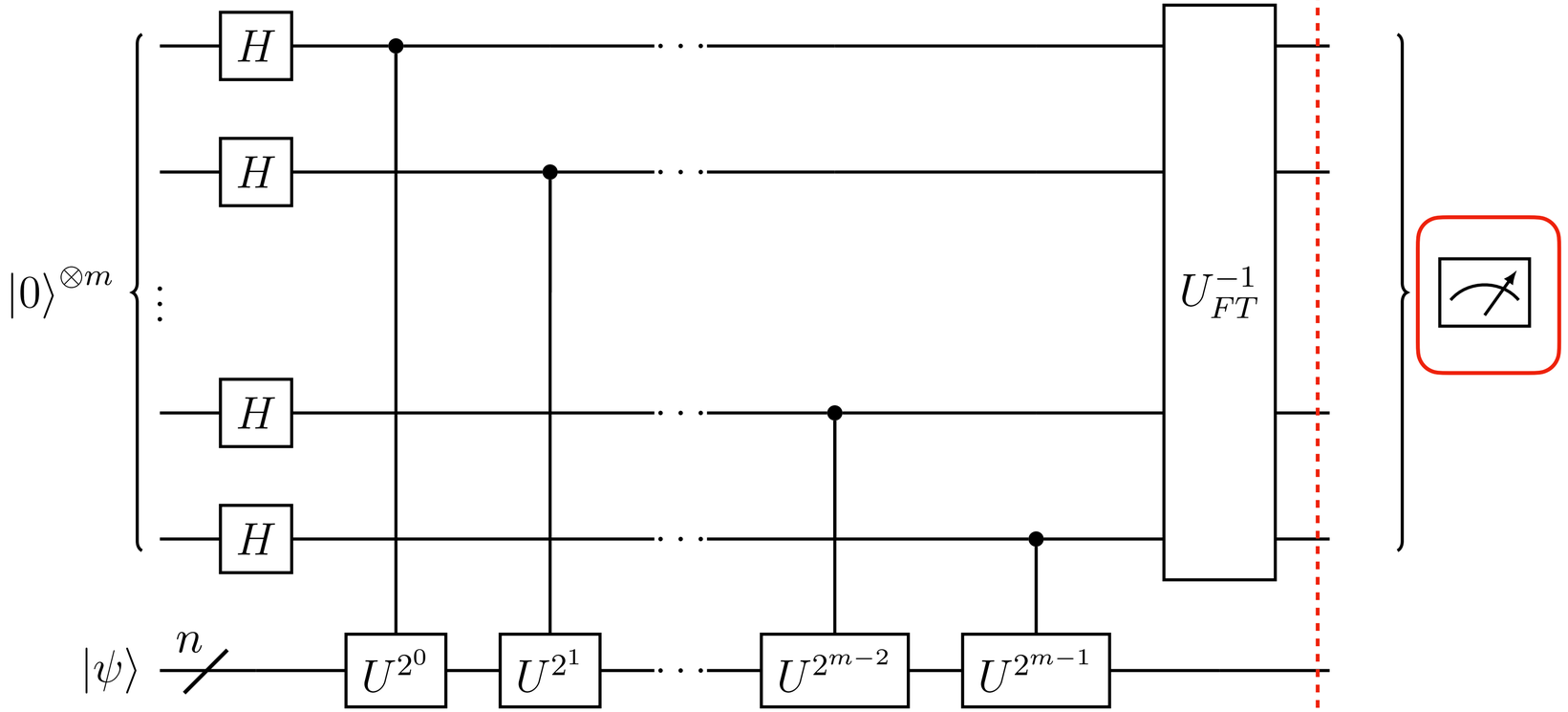
$p(1) = |\frac{1}{2}\left(1-e^{i\theta}\right)|^2 = \sin^2(\theta/2)$

😀

But need many "shots" (trials) to get good precision ☹️

Better: use more ancillary qubits! [Kitaev]

QPE



I: $|\psi\rangle|\dfrac{2^m\theta}{2\pi}\rangle_m$

Measurement gives integer closest to $\dfrac{2^m\theta}{2\pi}$

increasing m $\Rightarrow$ increasing precision of estimate of θ

Remarks

- If $|\psi\rangle$ is not an eigenstate of U, then the measurement *projects* $|\psi\rangle$ to an eigenstate of U
- Can regard Shor's algorithm as QPE with a particular choice of U

3. Towards physics applications

Naive observation:

Scaling of resources needed to simulate a spin-1/2 chain with L spins (qubits):

- Classical computer: exponential in L (Hilbert space dimension is 2^L)
- Quantum computer: linear in L potential *exponential* advantage!

… but it's not so simple

- Quantum state preparation

How to prepare a given quantum state (say, an eigenstate of a Hamiltonian) on a quantum computer?

Applications: computing correlation functions, entanglement entropy, etc. in this state

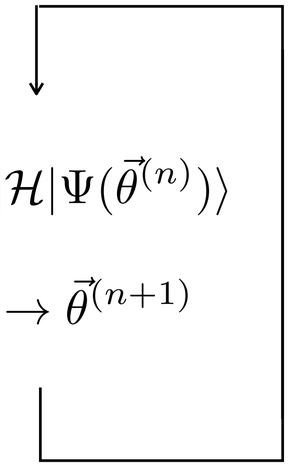
- next lecture!

- Variational Quantum Eigensolver (VQE)

Hybrid quantum/classical algorithm for estimating the ground-state energy E_0 of a Hamiltonian $\mathcal{H}$ using the variational theorem

normalized trial state $|\Psi(\vec{\theta})\rangle$ $\vec{\theta}$ parameters

iteration: $\vec{\theta}^{(0)}$



quantum $\langle\Psi(\vec{\theta}^{(n)})|\mathcal{H}|\Psi(\vec{\theta}^{(n)})\rangle$

classical $\vec{\theta}^{(n)} \rightarrow \vec{\theta}^{(n+1)}$

variational theorem $\Rightarrow$ $\langle\mathcal{H}\rangle \geq E_0$

challenge: find suitable trial state

- Quantum dynamics

Time evolution of a quantum system

$$|\psi(t)\rangle = e^{-i\mathcal{H}t}|\psi(0)\rangle$$

(assume Hamiltonian $\mathcal{H}$ is time-independent)

How to implement on a quantum computer?

- Suzuki-Trotter formulas
- Linear combination of unitaries/ Taylor expansion
- "Integrable Trotterization"

⋮

Suzuki-Trotter

- Divide evolution into time slices

$$e^{-i\mathcal{H}t} = \left(e^{-i\mathcal{H}t/n}\right)^n = \left(e^{-i\mathcal{H}\tau}\right)^n , \quad \tau = t/n$$

- $\mathcal{H} = \sum_j h_j$

$$e^{-i\mathcal{H}\tau} = e^{-i\tau\left(\sum_j h_j\right)} \approx \prod_j e^{-i\tau h_j}$$

first-order Suzuki-Trotter

error ~ $\mathcal{O}(\tau^2)$

•
•
•

Linear combination of unitaries/ Taylor expansion

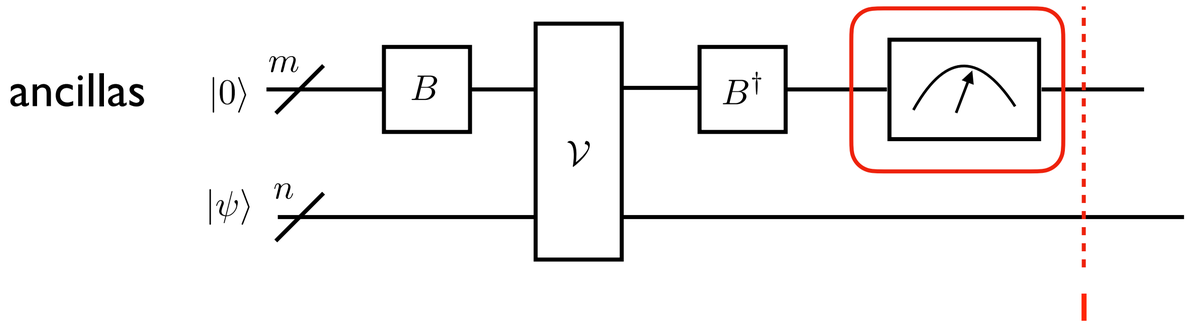
- Suppose $\mathcal{H} = \sum_j \alpha_j h_j$ h_j 's unitary $(*)$
- Divide evolution into time slices $e^{-i\mathcal{H}t} = \left(e^{-i\mathcal{H}t/n}\right)^n = \left(e^{-i\mathcal{H}\tau}\right)^n , \quad \tau = t/n$
- *Taylor expand* $e^{-i\mathcal{H}\tau} \approx \sum_k^K \frac{1}{k!} \left(-i\tau\mathcal{H}\right)^k$

$(*) \quad \Rightarrow \quad e^{-i\mathcal{H}\tau} \approx \sum_k^K \sum_{j_1,\ldots,j_k} \frac{1}{k!} \left(-i\tau\right)^k \alpha_{j_1} \cdots \alpha_{j_k} h_{j_1} \cdots h_{j_k}$

Linear combination of unitaries $U = \sum_x \beta_x V_x$

Introduce ancilla qubits, and prepare in state $B|0\rangle = \frac{1}{\sqrt{s}} \sum_x \sqrt{\beta_x}|x\rangle , \quad s = \sum_x \beta_x$

Define $\mathcal{V}\left(|x\rangle|\psi\rangle\right) = |x\rangle V_x|\psi\rangle$



IF measurement gives $|0\rangle_m$, then

I: $U|\psi\rangle_n|0\rangle_m$

probabilistic

"Integrable Trotterization"

[Vanicat, Zadnik, Prosen 2017; Destri, DeVega 1987, …]

For integrable Hamiltonian, say, XXX:

$$\mathcal{H} = \tfrac{1}{4}\sum_{j=1}^{L-1}(\vec{\sigma}_j \cdot \vec{\sigma}_{j+1} - \mathbb{I})$$

- Divide evolution into time slices

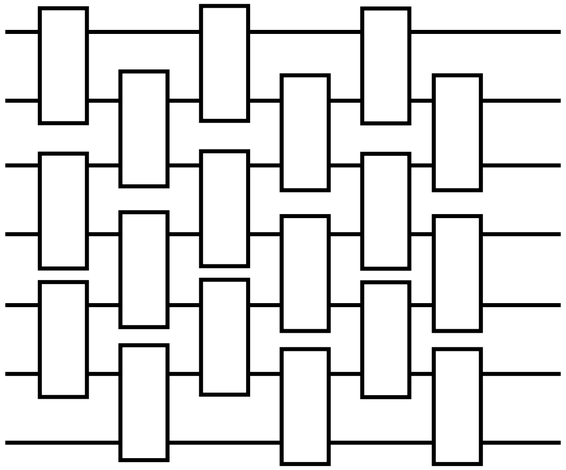
$$e^{-i\mathcal{H}t} = \left(e^{-i\mathcal{H}t/n}\right)^n = \left(e^{-i\mathcal{H}\tau}\right)^n, \quad \tau = t/n$$

- $e^{-i\mathcal{H}\tau} \approx U_e(\tau)\, U_o(\tau)$

$$U_o(\tau) = \prod_{j=1}^{L/2} V_{2j,2j+1}(\tau), \quad U_e(\tau) = \prod_{j=1}^{L/2} V_{2j-1,2j}(\tau)$$

"brickwork" circuit

$$V_{j,j+1}(\tau) = e^{-\frac{i\tau}{4}(\vec{\sigma}_j\cdot\vec{\sigma}_{j+1}-\mathbb{I})} = \check{R}(-\tan(\tau/2)), \quad \check{R}(u) = \frac{\mathbb{I}+iu\mathcal{P}}{1+iu}$$



time

4. Conclusions

- Quantum mechanics can be exploited for computing
- We are currently in the NISQ era; quantum computers are advancing rapidly
- Quantum computers may have useful physics applications

Thank you for your attention!