Quantum algorithms

Andrew Childs
University of Maryland

JOINT CENTER FOR
QUANTUM INFORMATION
AND COMPUTER SCIENCE

Overview

0. Introduction
1. Quantum query complexity
2. Algebraic problems
3. Quantum walk
4. Hamiltonian simulation
5. Quantum linear algebra
6. Optimization
7. Machine learning

0. Introduction

The origin of quantum speedup

Quantum computers allow for *interference between computational paths*

To perform a computation, we should arrange that

- paths to the solution interfere constructively
- paths to non-solutions interfere destructively

Quantum mechanics gives an *efficient representation of high-dimensional interference*

Quantum computing ≠ exponential parallelism

Can we just explore all potential solutions in parallel and pick out the correct one?

No! The linearity of quantum mechanics prohibits this.

To get significant speedup, quantum computers need to exploit structure

Key question: What kinds of problems have the right structure for quantum computers to exploit?

Unstructured search

Can quantum computers speed up brute-force search?

Given a black-box function $f : \{1, \ldots, N\} \to \{0, 1\}$, is there an $i \in \{1, \ldots, N\}$ such that $f(i) = 1$?

Classically: $\Theta(N)$ queries

Quantumly: $O(\sqrt{N})$ queries [Grover 96]

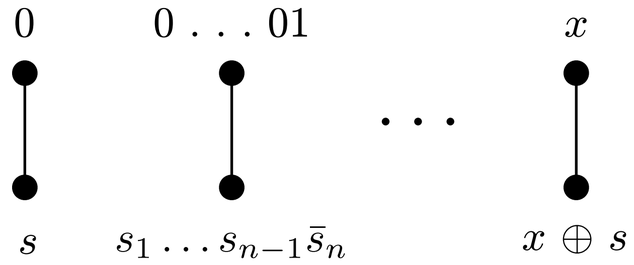
- by phase kickback, can implement an oracle $|i\rangle \mapsto (-1)^{f(i)}|i\rangle$
- this is a reflection about the M marked items
- alternate with reflection about $\frac{1}{\sqrt{N}} \sum_{i=1}^{N} |i\rangle$
- rotation by an angle $\Theta(1/\sqrt{N})$ in a 2D subspace
- significant overlap with marked subspace in time $O(\sqrt{N/M})$

Also quantumly: $\Omega(\sqrt{N})$ queries necessary [Bennett, Bernstein, Brassard, Vazirani 97]

Simon's problem

Given a black-box function $f : \{0,1\}^n \to R$

Promise: There is some $s \in \{0,1\}^n$ such that $f(x) = f(y)$ if and only if $x = y$ or $x = y \oplus s$



Problem: Find s

One classical strategy:

- Compute $f(x)$ for a random x
- Repeat until we find $x_i \neq x_j$ such that $f(x_i) = f(x_j)$
- Output $s = x_i \oplus x_j$

By the birthday problem, we need about $\sqrt{2^n}$ steps. This is essentially optimal.

Simon's algorithm

On a quantum computer, the ability to compute $f(x)$ corresponds to the ability to perform the unitary transformation $|x, 0\rangle \mapsto |x, f(x)\rangle$

Subroutine:

- Prepare the uniform superposition $\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, 0\rangle$
- Compute f in superposition $\mapsto \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, f(x)\rangle$
- Perform the Hadamard transform on the first n qubits
- Measure the state of the first n qubits

Fact: Measurement returns a uniformly random x subject to the condition $x \cdot s = 0$

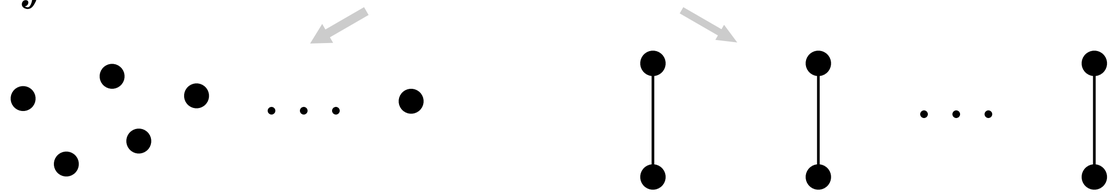
With $O(n)$ samples, these values determine s with good probability

Recall: $\Omega(\sqrt{2^n})$ classical queries. Exponential quantum speedup!

The collision problem

Given a black-box function $f : \{0,1\}^n \to R$ $\qquad N = 2^n$

Promise: f is either 1-to-1 or 2-to-1



Problem: Determine which holds

Can be solved with $O(N^{1/3})$ queries [Brassard, Høyer, Tapp 97]

- query K items
- search through remaining items for a duplicate
- cost $O(K + \sqrt{N/K})$ is minimized with $K = \Theta(N^{1/3})$

This is optimal! No exponential speedup. [Aaronson, Shi 01]

The prospect of quantum speedup

The collision problem does not have enough structure to allow a fast quantum algorithm

Simon's problem is a special case with enough additional structure to give a fast quantum algorithm (but not a fast classical algorithm) → exponential speedup

Major questions: What problems have fast quantum algorithms?

What structures enable exponential speedup?

Another important question: When can we get polynomial quantum speedup, and how much is possible?

1. Quantum query complexity

Quantum query model

Given a black box for an input string $x \in \Sigma^n$

- A query reveals $x_i \in \Sigma$ for any specified $i \in \{1, \ldots, n\}$
- A quantum query is the unitary operation $|i, z\rangle \mapsto |i, z + x_i\rangle$
  (This is the standard reversible computation of x_i; it can be done efficiently if we have an efficient circuit to compute x_i from i.)

Main question: How many queries are needed to compute some $f : P \to T$, where $P \subseteq \Sigma^n$ specifies a promise on the input?

Models:

- deterministic, $D(f)$: classical algorithm that always suceeds
- randomized, $R(f)$: randomized classical algorithm, success probability at least 2/3
- quantum, $Q(f)$: quantum algorithm, success probability at least 2/3

Query complexity is a *very clean setting* in which *lower bounds are feasible*.

Adversary method

The quantum adversary method [Ambainis 00; Høyer, Lee, Špalek 07] uses a progress measure that quantifies entanglement with an adversary who holds a superposition of instances.

Theorem. $Q(f) = \Omega(\mathrm{Adv}(f))$ where $\mathrm{Adv}(f) = \max_{\Gamma} \dfrac{\|\Gamma\|}{\max_i \|\Gamma_i\|}$

$$\Gamma_{xy} = 0 \text{ if } f(x) = f(y)$$

$$(\Gamma_i)_{xy} = \begin{cases} \Gamma_{xy} & \text{if } x_i \neq y_i \\ 0 & \text{if } x_i = y_i \end{cases}$$

Can be computed by a semidefinite program

In principle, always gives a tight bound (more later)! But can be hard to evaluate.

Some variants are easier to apply, but not necessarily tight.

Polynomial method

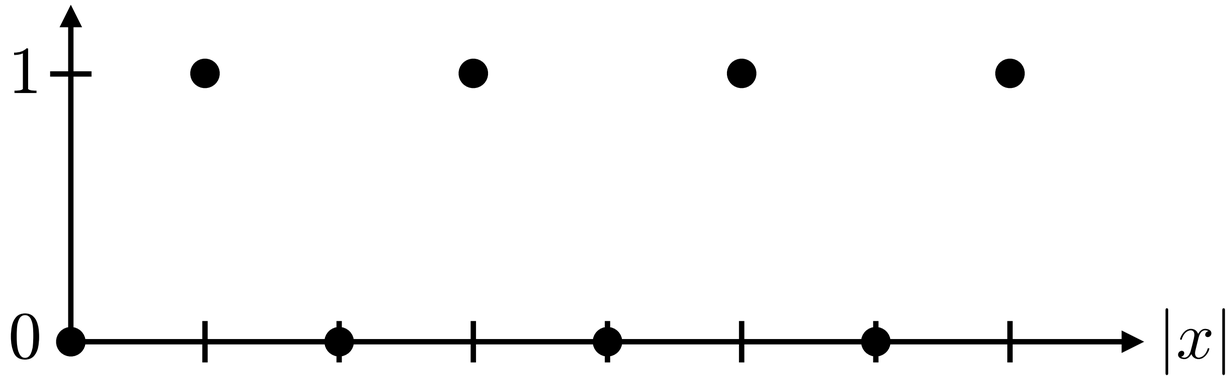
Another lower bound method uses a connection between quantum query algorithms and polynomials.

Lemma. The amplitudes of the final state of a t-query quantum algorithm with input $x \in \{0,1\}^n$ are polynomials in $x_1, \ldots, x_n$ of degree t.

Proof: Query $|i,z\rangle \mapsto |i, z \oplus x_i\rangle = (1-x_i)|i,z\rangle + x_i|i,\bar{z}\rangle$ increases degree by 1

So if we need a high-degree polynomial to represent the output, the query complexity must be high.

Example: Parity



Quantum speedup needs structure

Recall main question: How many queries are needed to compute $f : P \to T$ (where $P \subseteq \Sigma^n$ is the promise)?

If f is total ($P = \Sigma^n$) then $D(f) = O(Q(f)^6)$. [Beals, Buhrman, Cleve, Mosca, de Wolf 01]

recently improved to $D(f) = O(Q(f)^4)$. [Aaronson, Ben-David, Kothari, Rao, Tal 20]

So promises are necessary for exponential quantum speedup.

Symmetry can also prevent speedup by making the promise too unstructured.

Theorem. If f is permutation-invariant then $R(f) = O(Q(f)^3)$.

[Chailloux 18; improves Aaronson, Ambainis 11]

What other symmetries prevent exponential quantum speedup?

Symmetries of (hyper)graphs [Ben-David, Childs, Gilyén, Kretschmer, Podder, Wang 20]

Structured queries

Can get a structured query problem by giving access to some underlying object in a variety of different ways.

Example 1: Bernstein-Vazirani problem. Hidden string $s \in \{0,1\}^n$. Oracle reveals $x \cdot s$ for any input vector $x \in \{0,1\}^n$. Results of 2^n possible queries are specified by only n bits. Learning s takes n classical queries but only 1 quantum query.

Example 2: Search with wildcards. Hidden string $s \in \{0,1\}^n$. Oracle takes input $x \in \{0,1,*\}^n$ and tells whether x matches s, where * matches either 0 or 1. Learning s takes $\Omega(n)$ classical queries, $\tilde{O}(\sqrt{n})$ quantum queries. [Ambainis, Montanaro 14]

More recent examples, some with exponential speedup:

- Graph connectivity with cut queries [Lee, Santha, Zhang 20]
- Graph properties with OR or PARITY queries [Montanaro, Shao 20]
- Linear algebra with matrix-vector queries [Childs, Hung, Li 21]

Maximal separations

What is the largest possible quantum vs. classical query separation?

"Forrelation": $O(1)$ quantum vs. $\tilde{\Omega}(\sqrt{n})$ classical [Aaronson, Ambainis 14]

Recently improved to $\lceil k/2 \rceil$ quantum queries, $\tilde{\Omega}(n^{1-1/k})$ classical queries
[Sherstov, Storozhenko, Wu 20; Bansal, Sinha 20]

Optimal since t quantum queries can be simulated with $O(n^{1-1/2t})$ randomized queries [Aaronson, Ambainis 14]

What is the largest possible separation for a *total* function?

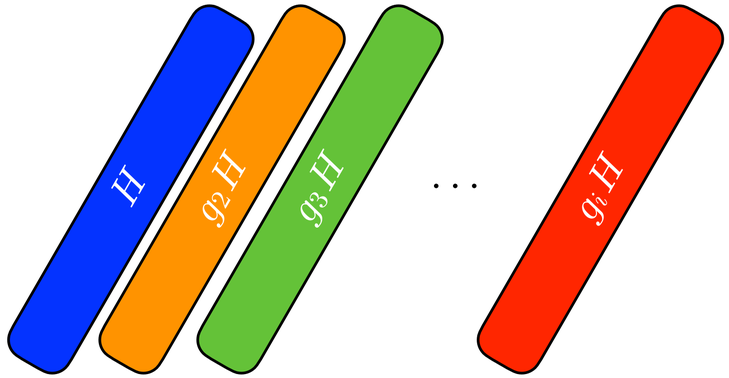
- Recall $R = O(Q^4)$ [Aaronson, Ben-David, Kothari, Rao, Tal 20]
- $R(\text{OR}) = \Omega(Q(\text{OR})^2)$
- $\exists f : \ R(f) = \Omega(Q(f)^{2.5})$ [Aaronson, Ben-David, Kothari 15]
- Exponent improved to 3 by the above papers [SSW 20; BS 20]

2. Algebraic problems

Hidden symmetry

Simon's problem exemplifies a more general class of problems with hidden symmetry

Hidden subgroup problem: Given a known group G and a black-box function $f : G \to R$. Promised that f is constant on cosets of some (unknown) subgroup $H \leq G$ and distinct on different cosets. Goal: find (a generating set for) H.



"Standard method": $|G\rangle := \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle \mapsto \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g, f(g)\rangle$

Discarding second register gives a *coset state* $|gH\rangle := \frac{1}{\sqrt{|H|}} \sum_{h \in H} |gh\rangle$ for a uniformly random (unknown) g

Finite abelian HSP

This problem can be solved efficiently whenever G is a finite abelian group.

Quantum Fourier transform over $\mathbb{Z}_N$:

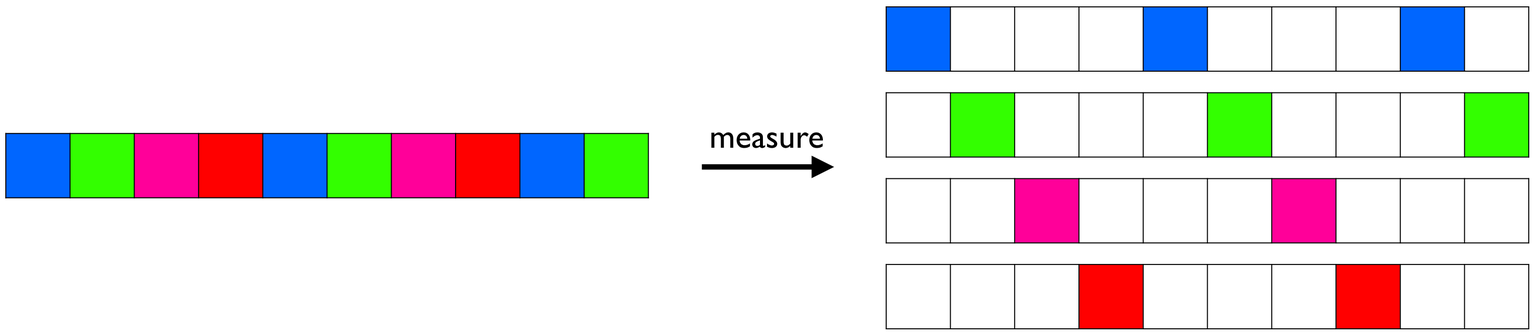
$$|x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k \in \mathbb{Z}_N} e^{2\pi i k x / N} |k\rangle$$

(generalizes to other groups)

Sampling from the a coset state in the Fourier basis gives a result that is "orthogonal to H" (more precisely, gives a character that is trivial on H). Polynomially many samples suffice to efficiently determine a generating set for H.

Infinite abelian HSPs

Shor's algorithm for factoring N finds the period of a function $f : \mathbb{Z} \to \mathbb{Z}_N$.
Can handle the non-finite domain by truncating to a sufficiently large subset.



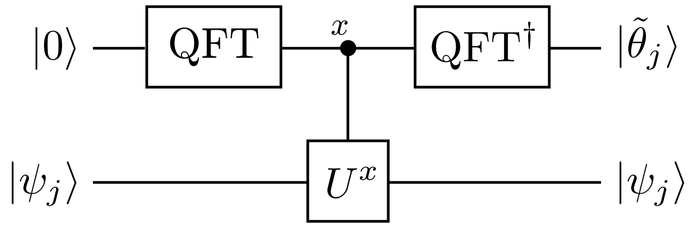
Algorithm for Pell's equation solves a hidden subgroup problem in $\mathbb{R}$ [Hallgren 02]

Algorithms for other problems in number fields handle HSPs in $\mathbb{R}^n$ [Hallgren 05; Hallgren, Eisenträger, Kitaev, Song 14]

Aside: Phase estimation

An equivalent approach to period finding is to apply *phase estimation* to a unitary operator that computes the periodic function (in place).

Problem: Given a unitary operator U with eigenvectors $|\psi_j\rangle$, where $U|\psi_j\rangle = e^{i\theta_j}|\psi_j\rangle$, produce an estimate of θ_j



$$\sum_j \alpha_j |0\rangle|\psi_j\rangle \mapsto \sum_j \alpha_j |\tilde{\theta}_j\rangle|\psi_j\rangle$$

To get an estimate with precision ϵ, we raise U to a power that is $O(1/\epsilon)$.

This is a useful tool in many quantum algorithms!

[Kitaev 95]

Abelian HSP applications

Finite abelian groups:

- Discrete log [Shor 94]
- Decomposing abelian groups [Cheung, Mosca 01]
- Counting points on curves [Kedlaya 06]

Infinite abelian groups:

- Factoring [Shor 94]
- Pell's equation ($x^2 - dy^2 = 1$) [Hallgren 02]
- Unit group of a number field [Hallgren 05; Hallgren, Eisenträger, Kitaev, Song 14]
- Principal ideal problem, class groups [Hallgren 05; Biasse, Song 16]
- Ray class groups, Hilbert class fields [Hallgren, Eisenträger 10]

Nonabelian HSP

What if G is a nonabelian group?

HSP definition still makes sense: given $f : G \to R$ constant on a subgroup $H \leq G$, distinct on different (say, left) cosets $H, g_1 H, g_2 H, \ldots$

The "standard method" generates coset states $|gH\rangle$ for uniformly random (unknown) $g \in G$

If H, K are distinct subgroups, the states $|xH\rangle$ and $|yK\rangle$ cannot have high overlap

This can be used to show that polynomially many coset state samples are sufficient to determine H [Ettinger, Hoyer, Knill 04]

The "only" problem: how do we determine H efficiently?

Nonabelian HSP: Examples and applications

Standard method algorithms can start with nonabelian Fourier sampling WLOG

Efficient algorithms known for specific HSPs: normal subgroups, metacyclic groups, Heisenberg/extraspecial groups, etc.

HSPs with exciting potential applications:
- Symmetric group: graph isomorphism, code equivalence
- Dihedral group: lattice problems [Regev 02]

A standard method algorithm for the symmetric group HSP would require highly entangled measurements [Hallgren, Moore, Rötteler, Russell, Sen, 06]

"Kuperberg sieve" solves the dihedral HSP in subexponential time
- No quantum speedup for lattice problems
- Subexponential quantum algorithm for elliptic curve isogenies [Childs, Jao, Soukharev 14]

Dihedral HSP challenge problem

The standard method and Fourier sampling produces a qubit state

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi isk/N}|1\rangle)$$

with $k \in \mathbb{Z}_N$ known, selected uniformly at random.

With $\mathrm{poly}(\log N)$ samples of such states, we have enough info to determine s.

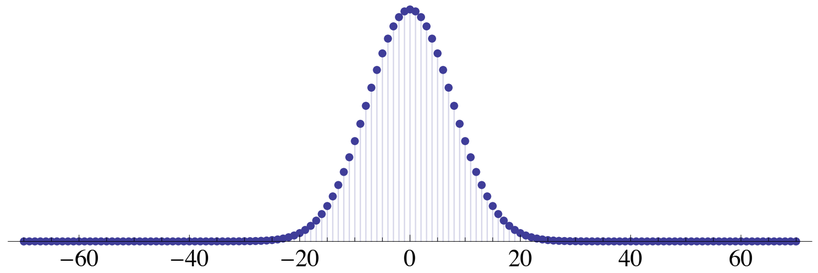
Can we determine s efficiently?

3. Quantum walk

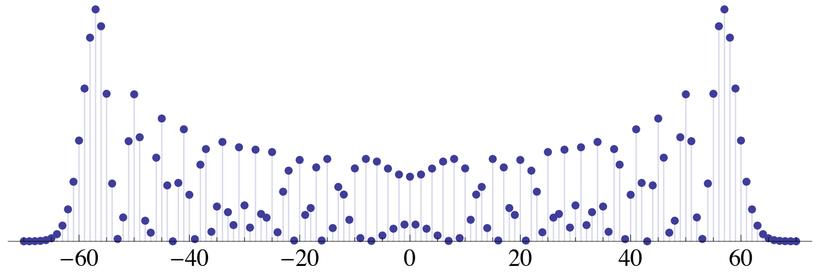
From random to quantum walk

Quantum analog of a random walk on a graph.

Idea: Replace probabilities by quantum amplitudes.
Interference can produce radically different behavior!



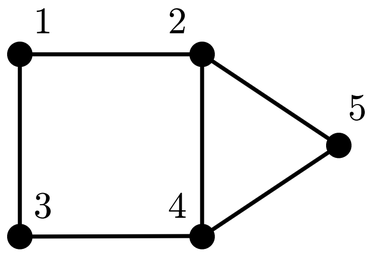
classical



quantum

Continuous-time quantum walk

Graph G:



$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

adjacency matrix

$$L = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 3 & 0 & -1 & -1 \\ -1 & 0 & 2 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{pmatrix}$$

Laplacian

Random walk on G

State: Probability $p_v(t)$ of being at vertex v at time t

Dynamics: $\dfrac{\mathrm{d}}{\mathrm{d}t}\vec{p} = L\vec{p}$

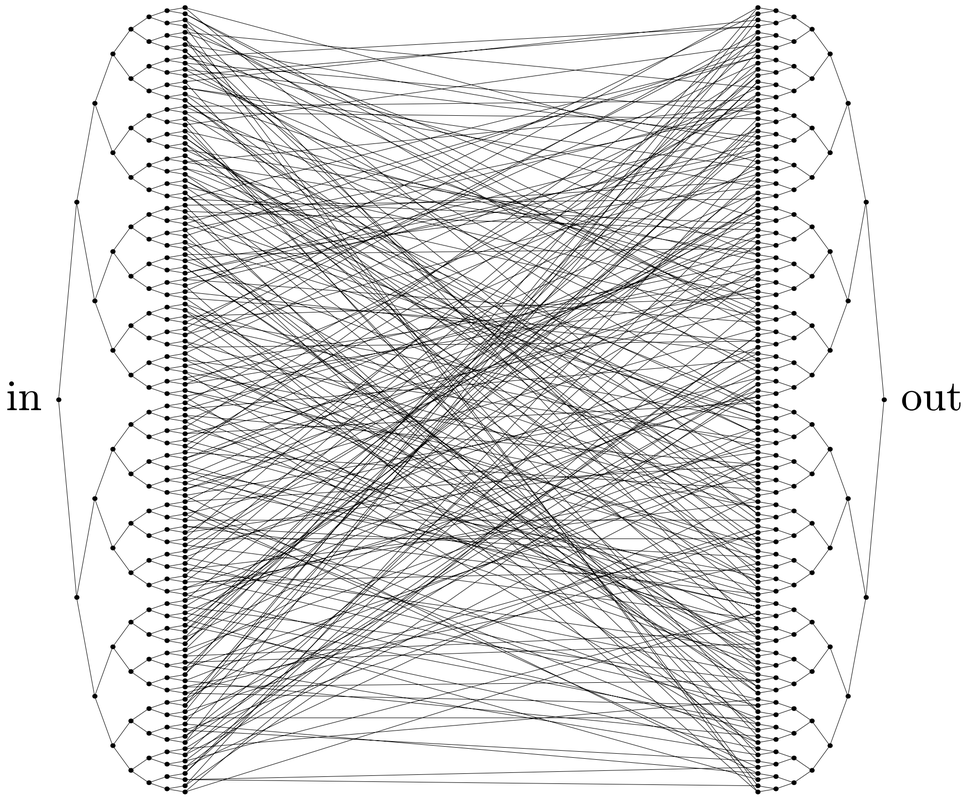
Quantum walk on G

State: Amplitude $a_v(t)$ to be at vertex v at time t

Dynamics: $i\dfrac{\mathrm{d}}{\mathrm{d}t}\vec{a} = L\vec{a}$

$$i\frac{\mathrm{d}}{\mathrm{d}t}\vec{a} = A\vec{a}$$

Exponential speedup



Problem: Given the label of in and an adjacency-list black box for the graph, find the label of out.

Quantum walk from $|\mathrm{in}\rangle$ stays in the *column subspace* (uniform superpositions over vertices at fixed distance from in).

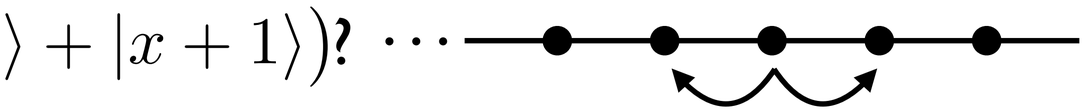
This walk rapidly reaches a state with significant overlap on $|\mathrm{out}\rangle$.

Using polynomially many queries, a classical algorithm cannot distinguish the graph from an infinite binary tree rooted at in.

[Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 03]

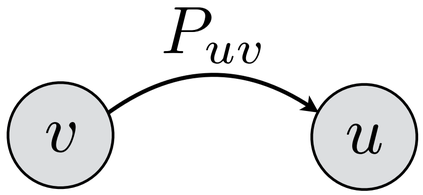
Discrete-time quantum walk

A walk with discrete time steps is a little harder to define.

On a path: $|x\rangle \mapsto \frac{1}{\sqrt{2}}\big(|x-1\rangle + |x+1\rangle\big)$?  Not unitary!

Solution: Introduce another register ("coin") that remembers the previous position
(reduces the potential for interference, but only slightly)

Szegedy walk: For a stochastic transition matrix P,



- Reflect about $\mathrm{span}\{|\psi_v\rangle : v \in V\}$
  where $|\psi_v\rangle := \sum_{u \in V} \sqrt{P_{uv}}|v,u\rangle$
- Swap the edge direction: $S := \sum_{u,v \in V} |u,v\rangle\langle v,u|$

[Szegedy 05]

Quantum walk search

Problem: Given a graph $G = (V, E)$ with a subset $M \subseteq V$ of *marked vertices*. Using an oracle that tells whether a vertex is marked, determine whether M is empty.

Classical strategy: Take a random walk until we reach a marked vertex.

Time to hit a marked vertex is $O(1/\delta\epsilon)$, where

δ = spectral gap of walk $\quad \epsilon = |M|/|V|$

(second-largest magnitude of an eigenvalue of transition matrix is $1 - \delta$)

Quantum strategy: Consider the Szegedization of the *absorbing walk* that remains at a marked vertex

Perform phase estimation on $|\psi\rangle \propto \sum_{x \notin M} |\psi_x\rangle$

This state is invariant if $|M| = 0$ and lives in eigenspaces with phase $\Omega(\sqrt{\delta\epsilon})$ if $|M| \neq 0$, so $O(1/\sqrt{\delta\epsilon})$ steps of the walk suffice to determine whether $|M| = 0$.

Quantum walk search: examples

Unstructured search: $G =$ complete graph on N vertices $\quad \delta = \Theta(1) \quad \epsilon = 1/N$

Classical: $O(N)$ $\quad$ Quantum: $O(\sqrt{N})$

Element distinctness: [Ambainis 04]

Given $f : [N] \to R$, are there distinct $x, y \in [N]$ with $f(x) = f(y)$? $\quad [N] := \{1, \ldots, N\}$

Classical: $\Omega(N)$

Quantum: Consider walk on Hamming graph $H(N, K)$

vertices = $[N]^K$, edges between K-tuples that differ in one coordinate

store function values associated with the K inputs

$\delta = \Omega(1/K) \quad \epsilon = \Omega((K/N)^2)$

complexity $K + N/\sqrt{K}$, optimized with $K = N^{2/3}$

This provides a powerful, general tool for search problems

Quantum walk search: refinements and generalization

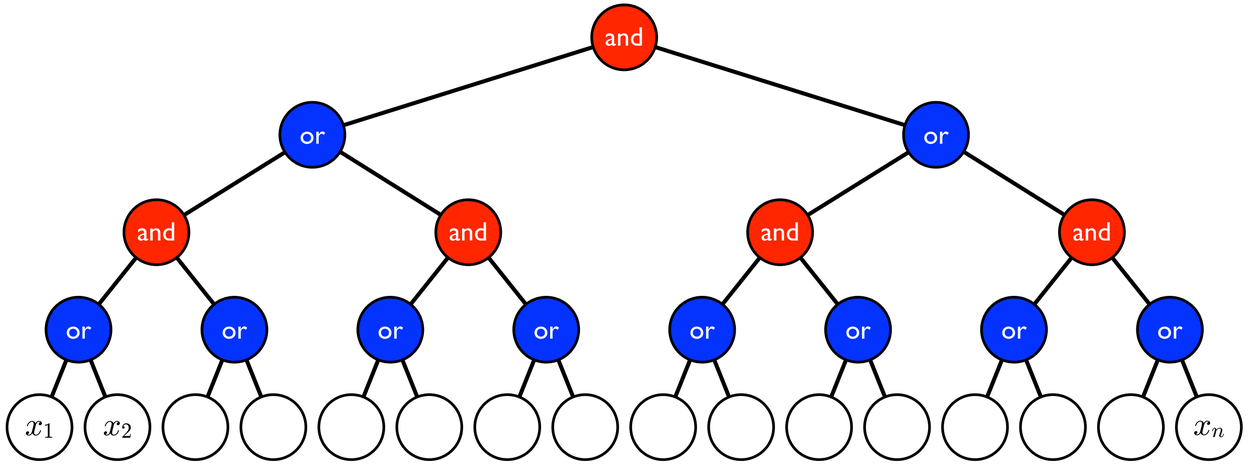
Can give a quantum walk search algorithm with quadratic speedup over the classical hitting time, not just the upper bound $O(1/\delta\epsilon)$ [Szegedy 05]

Quantum walk can find (multiple) marked items in this time

[Ambainis, Gilyén, Jeffery, Kokainis 20]

Formula evaluation

Consider a balanced binary AND-OR tree:

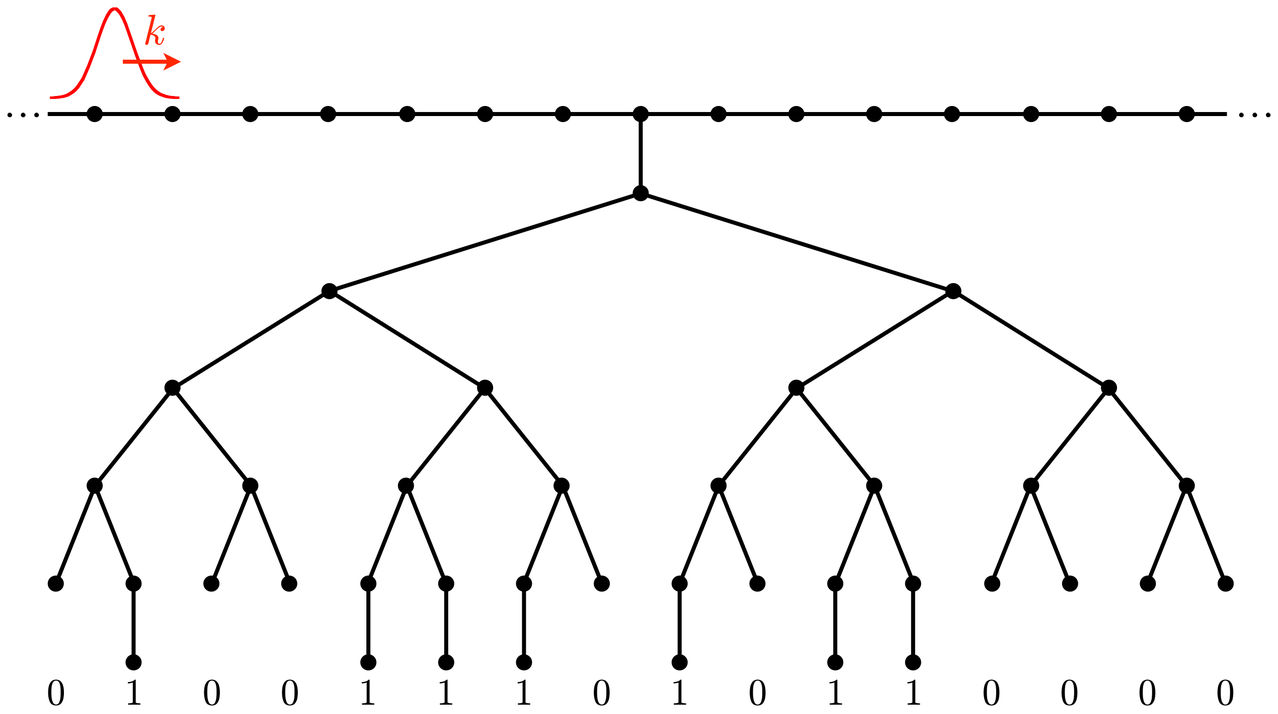


Classical complexity: $\Theta(n^{0.753\ldots})$ [Snir 85; Saks, Wigderson 86; Santha 95]

Quantum lower bound: $\Omega(\sqrt{n})$ [Barnum, Saks 02]

(holds for arbitrary AND-OR formulas)

Formula evaluation by scattering



Claim: For $k = \Theta(1/\sqrt{n})$, the wave is transmitted if the formula (translated into NAND gates) evaluates to 0, and reflected if it evaluates to 1. [Farhi, Goldstone, Gutmann 07]

General formulas and span programs

In fact the quantum query complexity of *any* n-input AND-OR formula is $O(\sqrt{n})$

[Reichardt 10]

One approach: apply phase estimation to a quantum walk on a tree that encodes the formula

Alternative: construct a *span program*, composing span programs for elementary gates

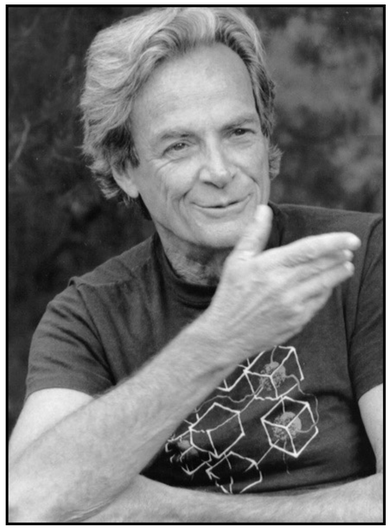
Recall the quantum adversary method: $\mathrm{Adv}(f) = \max_{\Gamma} \frac{\|\Gamma\|}{\max_i \|\Gamma_i\|}$

The dual of this semidefinite program can be used to construct a quantum algorithm for evaluating f with $O(\mathrm{Adv}(f))$ queries (apply phase estimation to a kind of generalized quantum walk)

Useful for understanding general features of query complexity.
In particular: $\mathrm{Adv}(f \circ g) \leq \mathrm{Adv}(f)\,\mathrm{Adv}(g)$

4. Hamiltonian simulation

Simulating Hamiltonian dynamics



"… 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)
Simulating physics with computers

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

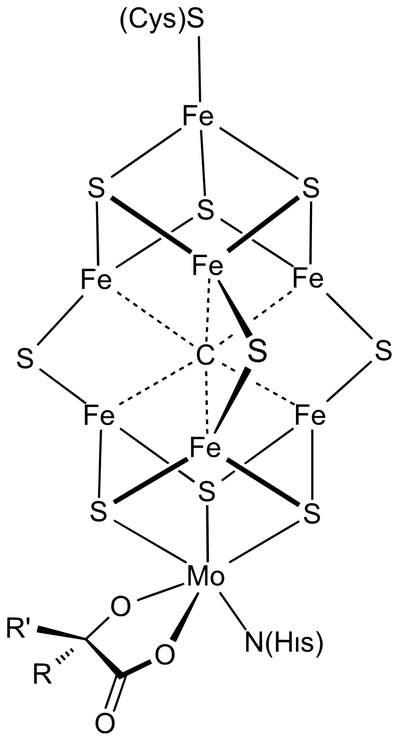
A classical computer cannot even represent the state efficiently.

A quantum computer cannot produce a complete description of the state.

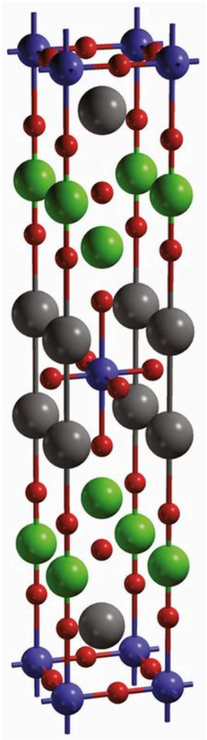
But given succinct descriptions of
- the initial state (suitable for a quantum computer to prepare it efficiently) and
- a final measurement (say, measurements of the individual qubits in some basis),

a quantum computer can efficiently answer questions that (apparently) a classical one cannot. Simulation is BQP-complete!

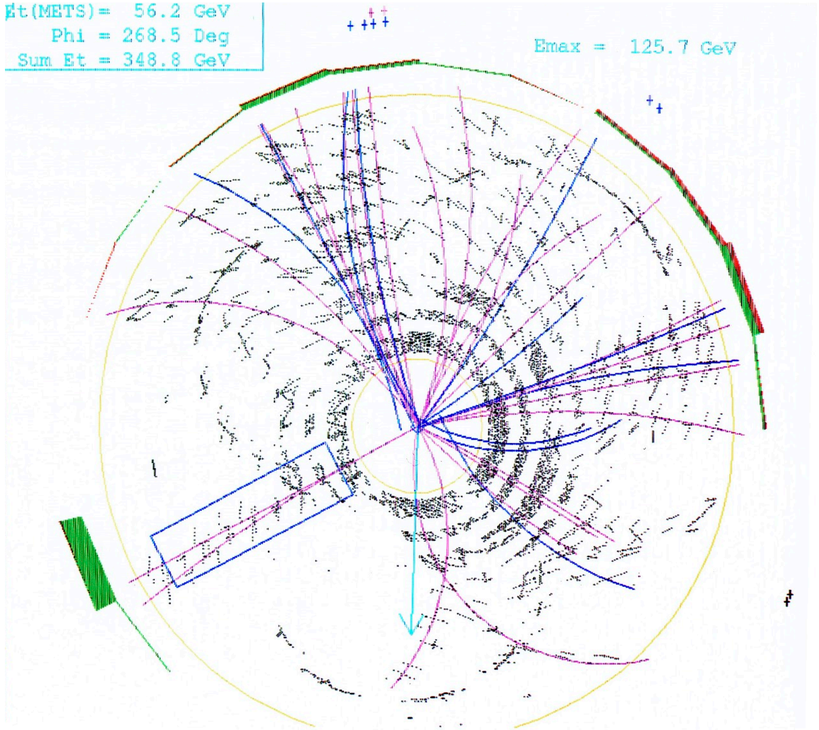
Computational quantum physics



quantum chemistry
(e.g., nitrogen fixation)

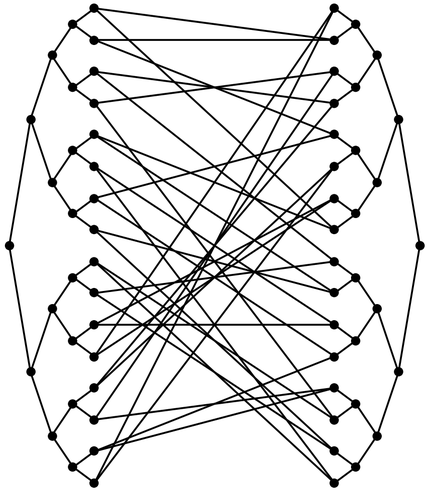


condensed matter physics/
properties of materials

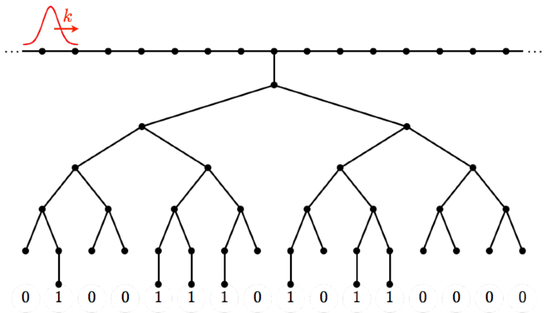


nuclear/particle
physics

Implementing quantum algorithms



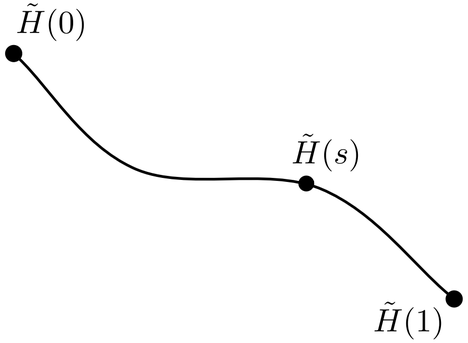
exponential speedup by quantum walk



evaluating Boolean formulas

$$A|x\rangle = |b\rangle$$

linear/ differential equations, convex optimization



adiabatic optimization

Product formulas

Suppose we want to simulate $H = \sum_{\ell=1}^{L} H_\ell$

Combine individual simulations with the Lie product formula. E.g., with two terms:

$$\lim_{r\to\infty} \left(e^{-iAt/r} e^{-iBt/r}\right)^r = e^{-i(A+B)t}$$

$$\left(e^{-iAt/r} e^{-iBt/r}\right)^r = e^{-i(A+B)t} + O(t^2/r)$$

To ensure error at most ϵ, take
$r = O\left((\|H\|t)^2/\epsilon\right)$ [Lloyd 96]

Gives simulation of d-sparse Hamiltonians with complexity $\mathrm{poly}(d)$ [Aharonov, Ta-Shma 03]

To get a better approximation, use higher-order formulas.

E.g., second order:

$$\left(e^{-iAt/2r} e^{-iBt} e^{-iAt/2r}\right)^r = e^{-i(A+B)t} + O(t^3/r^2)$$

Systematic expansions to arbitrary order are known [Suzuki 92]

Using the $2k$th order expansion, the number of exponentials required for an approximation with error at most ϵ is at most

$$5^{2k} L^2 \|H\| t \left(\frac{L\|H\|t}{\epsilon}\right)^{1/2k}$$

[Berry, Ahokas, Cleve, Sanders 07]

Post-Trotter algorithms I

Linear-time simulation

"No Fast-Fowarding Theorem": simulation for time t has complexity $\Omega(t)$

[Berry, Ahokas, Cleve, Sanders 07]

Applying phase estimation to a Szegedization of H gives an $O(t)$ simulation

[Childs10; Berry, Childs 12]

High-precision simulation

Directly implement the truncated Taylor series of $\exp(-iHt)$, cost $O\left(t\frac{\log(t/\epsilon)}{\log\log(t/\epsilon)}\right)$

LCU Lemma: implement $U = \sum_j \beta_j V_j$ with complexity $O(\sum_j |\beta_j|)$

This is the optimal dependence on ϵ

[Berry, Childs, Cleve, Kothari, Somma 14 & 15]

Post-Trotter algorithms II

Optimal tradeoff

- Quantum signal processing (QSP) implements polynomials of a given "block-encoded" Hamiltonian (or more general matrix) $U = \begin{pmatrix} H & \cdot \\ \cdot & \cdot \end{pmatrix}$
- Gives d-sparse Hamiltonian simulation with cost $O(dt + \log(1/\epsilon))$ [Low, Chuang 17]
- QSP and "quantum singular value transformation" [Gilyén, Su, Low, Wiebe 19] provide versatile tools for other tasks

Lattice Hamiltonians

- Can do even better if the Hamiltonian has spatially local interactions
- All above methods use $\Omega(n^2)$ gates to simulate n spins with local interactions for constant time
- Combining forward and backward evolution and applying Lieb-Robinson bounds, can improve this to $\tilde{O}(n)$, which is optimal [Haah, Hastings, Kothari, Low 18]

Also other algorithms using multiproduct formulas, interaction picture, randomization, other norms, …

Product formulas strike back

Numerical simulations suggest that product formulas can perform much better than straightforward bounds show

Can give tighter bounds using integral representations of the error

$$e^{-iBt}e^{-iAt} - e^{-i(A+B)t} = \int_0^t \mathrm{d}\tau_1 \int_0^{\tau_1} \mathrm{d}\tau_2\, e^{-i(A+B)(t-\tau_1)} e^{i(\tau_2-\tau_1)B} [A,B] e^{-i\tau_2 B} e^{-i\tau_1 A}$$

Provides bounds that can take advantage of small commutators between terms

In particular, shows that product formulas nearly reproduce the complexity of [Haah, Hastings, Kothari, Low 18] for lattice Hamiltonians [Childs, Su, Tran, Wiebe, Zhu 19]

Can give even better bounds if we know the state has low energy [Sahinoglu, Somma 20]

Quantum chemistry

Algorithms depend on many choices:

- Often assume nuclei at fixed positions (Born-Oppenheimer approximation)
- Choose a set of electron basis functions (molecular orbitals, plane waves, etc.)

$$H = \sum_{ij} h_{ij} a_i^\dagger a_j + \sum_{ijkl} g_{ijkl} a_i^\dagger a_j^\dagger a_k a_l \qquad \text{fermion operators: } \{a_i, a_j\} = 0,\ \{a_i, a_j^\dagger\} = \delta_{ij}$$

- Convert to spins using a suitable transformation (Jordan-Wigner, Bravyi-Kitaev, etc.)

- Represent in first (locations of electrons) or second (occupation of modes) quantization

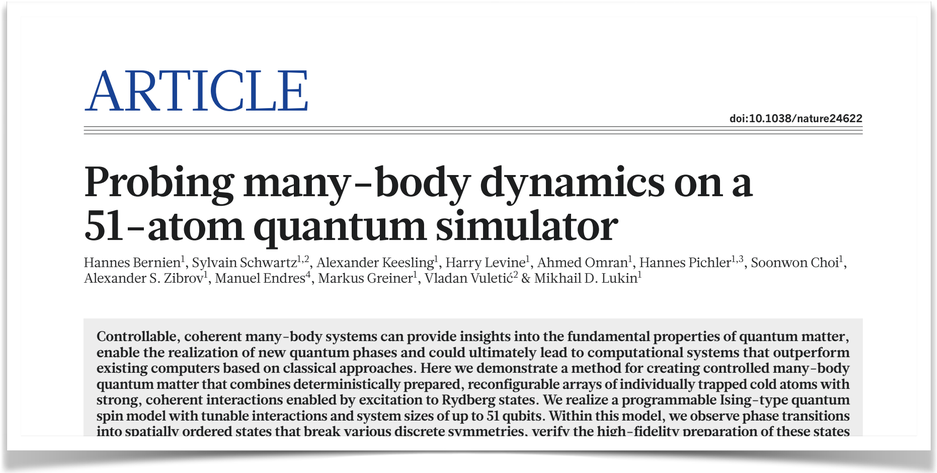
Selected asymptotic complexities (N modes, η electrons):

- [Wecker, Bauer, Clark, Hastings, Troyer 14] (2nd quantization, any basis): $O(N^{10})$
- [Babbush, Berry, Kivlichan, Wei, Love, Aspuru-Guzik 16] (2nd quantization, any basis): $O(N^5)$
- [Low, Wiebe 18] (2nd quantization, plane waves): $O(N^2)$
- [Babbush, Berry, McClean, Neven 18] (1st quantization, plane waves): $O(N^{1/3}\eta^{8/3})$

Analog simulation

Another approach: Construct a system that is described by the Hamiltonian you want to understand, and let it evolve!

Experimental efforts are further along than digital simulators



ARTICLE

doi:10.1038/nature24622

Probing many-body dynamics on a 51-atom quantum simulator

Hannes Bernien[1], Sylvain Schwartz[1,2], Alexander Keesling[1], Harry Levine[1], Ahmed Omran[1], Hannes Pichler[1,3], Soonwon Choi[1], Alexander S. Zibrov[1], Manuel Endres[4], Markus Greiner[1], Vladan Vuletić[2] & Mikhail D. Lukin[1]

Controllable, coherent many-body systems can provide insights into the fundamental properties of quantum matter, enable the realization of new quantum phases and could ultimately lead to computational systems that outperform existing computers based on classical approaches. Here we demonstrate a method for creating controlled many-body quantum matter that combines deterministically prepared, reconfigurable arrays of individually trapped cold atoms with strong, coherent interactions enabled by excitation to Rydberg states. We realize a programmable Ising-type quantum spin model with tunable interactions and system sizes of up to 51 qubits. Within this model, we observe phase transitions into spatially ordered states that break various discrete symmetries, verify the high-fidelity preparation of these states

Key questions:

- What kind of control is needed to realize Hamiltonians of interest?
- How can we be confident in the results?

Cubitt, Montanaro, Piddock 18: Universality result for spin models on lattices

Zhou, Aharonov 21: Universality that does not require spatial locality of the target Hamiltonian

5. Quantum linear algebra

Quantum linear systems algorithm

Given an $N \times N$ system of linear equations $Ax = b$, find $x = A^{-1}b$

Classical (or quantum!) algorithms need time $\Omega(N)$ just to write down x

What if we change the model?

- A is sparse; given a black box that specifies the nonzero entries in any given row or column
- Can efficiently prepare a quantum state $|b\rangle$
- Goal is to prepare a state $|x\rangle \propto A^{-1}|b\rangle$

We can do this in time $\mathrm{poly}(\log N, 1/\epsilon, \kappa)$ where $\kappa := \|A\| \cdot \|A^{-1}\|$

[Harrow, Hassidim, Lloyd 09]

Algorithm estimates the eigenvalues of A (in superposition) and replaces them by their inverse (using postselection)

Subsequent improvements do the same with complexity $\kappa\,\mathrm{poly}(\log(1/\epsilon))$ using variable-time amplitude amplification and LCU [Ambainis 12; Childs, Kothari, Somma 17]

Differential equations

We can apply a similar framework to other linear-algebraic tasks. For example:

Given a system of linear differential equations $\frac{\mathrm{d}}{\mathrm{d}t}x = Ax + b$
with the ability to prepare $|b\rangle$ and $|x(0)\rangle$, and a sparse matrix oracle for A,
prepare $|x(T)\rangle$ for some desired final time T

Approach: apply a finite difference approximation to give a linear system; solve it with the QLSA [Berry 14]

Generalizations give improved performance and also handle time-dependent coefficients, partial differential equations, some nonlinear differential equations, …

Applications?

Linear equations and differential equations are ubiquitous. Surely we can use this for something?

Proposals: electromagnetic scattering, machine learning, finance, …

The input/output requirements impose serious constraints.
No compelling end-to-end application with rigorous evidence for speedup.

Explicit output

Suppose we are given adjacency-list query access to A and an explicit vector b.
Can we find an explicit description of $x = A^{-1}b$ with (polynomial) quantum speedup?

Yes, if A is the Laplacian of a (weighted) graph, or more generally, if it is symmetric and weakly diagonally dominant. [Apers, Lee, de Wolf 20]

Related to the problem of "spectral sparsification."

Further polynomial speedups for quantum linear algebra?

6. Optimization

Discrete optimization

Grover's algorithm ⇒ quadratic speedup for minimization [Dürr, Hoyer 96]

Graph algorithms
- shortest paths [Dürr, Heiligman, Høyer, Mhalla 04]
- minimum spanning trees [Dürr, Heiligman, Høyer, Mhalla 04]
- maximum flows/matchings [Ambainis, Špalek 07]

Speeding up exponential-time algorithms for NP-hard problems (SAT, subset sum, lattice problems, TSP, set cover, …)

Some of these algorithms introduce interesting new tools:
- quantum backtracking using quantum walk [Montanaro 16]
- quantum methods for dynamic programming [Ambainis, Balodis, Iraids, Kokainis, Prusis, Vihrovs 18]

Continuous optimization

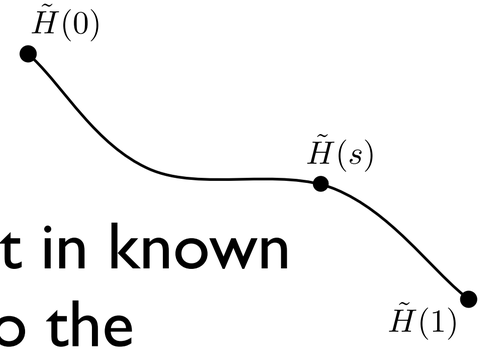
Linear/semidefinite programming

- polynomial speedups based on Gibbs sampling [Brandão, Svore 17; van Apeldoorn, Gilyén 19]
- faster algorithms in a stronger input model [Brandão, Kalev, Li, Lin, Svore, Wu 19]

Gradient-based algorithms

- Fast algorithm for computing gradients [Jordan 05]
- Minimization using gradient descent [Rebentrost, Schuld, Wossnig, Petruccione, Lloyd 19; Kerenidis, Prakash 20]
- Quantum query speedup for convex optimization with membership and evaluation oracles [van Apeldoorn, Gilyén, Gribling, de Wolf 20; Chakrabarti, Childs, Li, Wu 20]
- For high-dimensional non-smooth convex optimization with a gradient oracle, cannot achieve a quantum speedup as a function of the allowed error [Garg, Kothari, Netrapalli, Sherif 20]

Adiabatic optimization and QAOA



Strategy: encode a constraint problem with a diagonal Hamiltonian. Start in known ground state of a simple, non-diagonal Hamiltonian. Slowly interpolate to the problem Hamiltonian to produce its ground state. [Farhi, Goldstone, Gutmann, Sipser 00]

Complexity depends on the minimum spectral gap, but this is hard to estimate.

Often this is done with a Hamiltonian that has all negative off-diagonal entries ("stoquastic"). Then we can in principle apply quantum Monte Carlo (a classical algorithm), but its efficiency is also unclear.

Related strategy: quantum approximate optimization algorithm (QAOA). Alternate between diagonal & off-diagonal evolutions with optimized parameters. [Farhi, Goldstone, Gutmann 14]

7. Machine learning

Quantum machine learning

A challenge: much of the impressive success of classical machine learning is empirical

Quantum algorithms for some ML tasks have been proposed, e.g., recommendation systems [Kerenidis, Prakash 17]

Data structures that enable coherent quantum access can be exploited classically [Tang 19]

Other proposed algorithms for principal component analysis, clustering, etc. Potential for quantum speedup is unclear.

Another direction: computational learning theory [survey: Arunachalam, de Wolf 17]

Learn a concept given the ability to interact with it quantumly

- query access to a concept $c\colon \{0,1\}^n \to \{0,1\}$
- quantum examples $\sum_x \sqrt{p_x}|x, c(x)\rangle$

Conclusion

Outlook

Finding quantum algorithms is hard!

- Quantum mechanics is nonintuitive
- Classical algorithms are powerful
- We have limited quantum techniques

But we have come a long way in the 25+ years since Shor's algorithm

- New exponential speedups
- New techniques
- Much better understanding of quantum query complexity

Large-scale quantum computers could dramatically change our understanding of quantum algorithms

Challenge problems

1. Quantum query complexity
   - triangle problem ($\Omega(n),\ O(n^{5/4})$ [Le Gall 14])
2. Algebraic problems
   - quantum algorithms/hardness results for lattice problems
3. Quantum walk
   - exponential speedups for natural problems
4. Hamiltonian simulation
   - more practical algorithms for quantum chemistry, other applications
   - theoretical foundations for robust analog simulation
5. Quantum linear algebra
   - end-to-end applications
6. Optimization
   - query complexity of convex optimization
   - better evidence for/against exponential speedup by adiabatic optimization/QAOA
7. Machine learning
   - evidence for speedup in a realistic model

Further reading

Quantum Algorithm Zoo: quantumalgorithmzoo.org

Lecture notes: cs.umd.edu/~amchilds/qa/

Montanaro survey: arXiv:1511.04206

András Gilyén tutorial (QIP 2020): www.koushare.com/video/videodetail/4073

Topical surveys:

- quantum walk search (Santha): arXiv:0808.0059
- quantum walk (Reitzner, Nagaj, Buzek): arXiv:1207.7283
- algebraic problems (Childs, van Dam): arXiv:0812.0380
- optimization (de Wolf): https://youtu.be/1-2LIopvNIk
- computational learning theory (Arunachalam, de Wolf): arXiv:1701.06806