Algorithms for quantum computers

Andrew Childs

CS, UMIACS, & QuICS
University of Maryland

Joint Center for Quantum Information and Computer Science

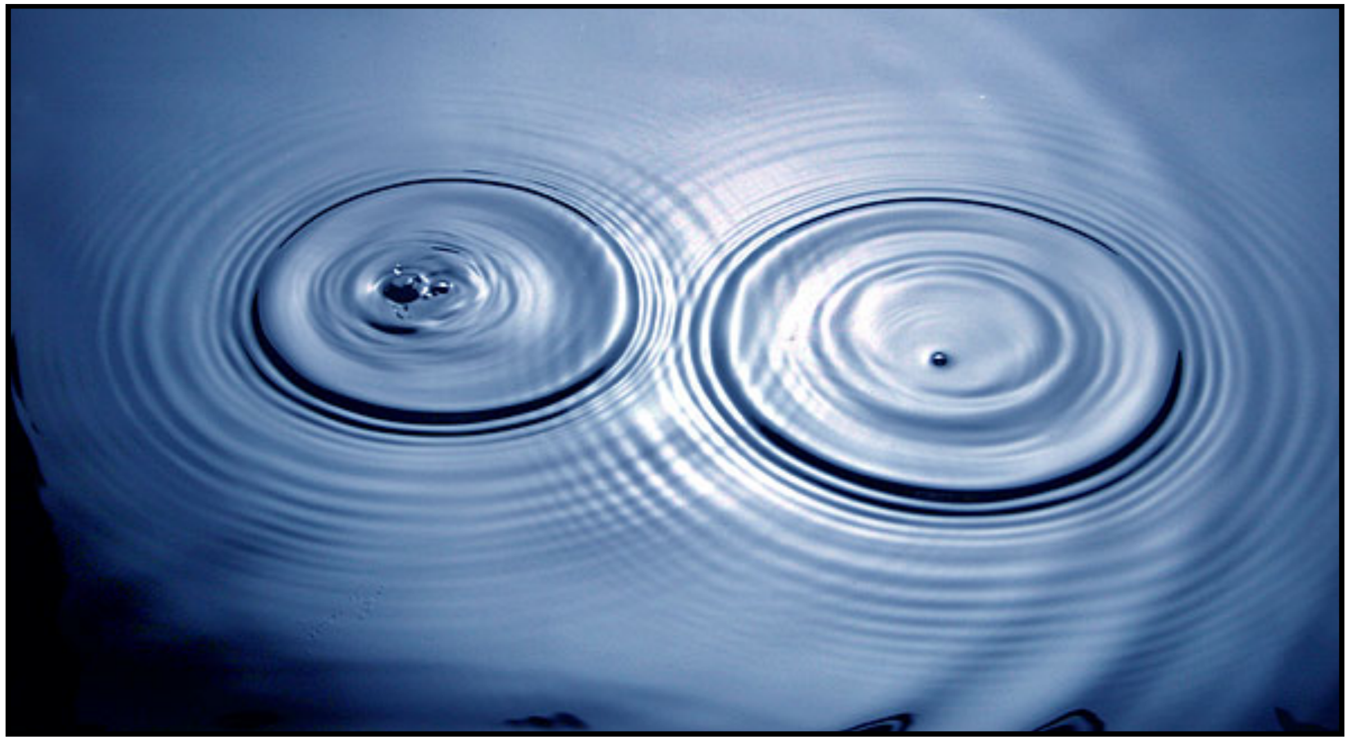
quics.umd.edu

Outline

0. The origin of quantum speedup
1. Hidden symmetries
2. Search
3. Optimization
4. Simulating quantum mechanics
5. Linear algebra in Hilbert space

The origin of quantum speedup

Interference between computational paths



Arrange so that

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

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

Hidden symmetries

Shor 1994: Efficient quantum algorithm for factoring integers
Widely believed to be classically hard RSA

Main idea: find period of $f(x) = a^x \bmod N$ for random a using the QFT, revealing factors of N

Related ideas lead to quantum algorithms for other problems: Computing discrete logarithms [Shor 94], decomposing abelian groups [Cheung, Mosca 01], algorithms for number fields [Hallgren 02 + more], counting points on algebraic curves [Kedlaya 06], attacks on symmetric crypto [Kuwakado, Morii 10; Kaplan et al. 16], ...

Nonabelian symmetries: Few algorithms but intriguing potential applications (symmetric group → graph isomorphism; dihedral group → lattice problems [Regev 04], elliptic curve isogenies [Childs, Jao, Soukharev 12]; general linear group → code equivalence)

Search

Grover 96: Unstructured combinatorial search over N possibilities using $O(\sqrt{N})$ queries (optimal)

Quantum analogs of random walks can sometimes explore graphs faster; quantum walk search sometimes achieves polynomial speedup over classical computation [Ambainis 03; Szegedy 04; Magniez et al. 06]

Applications: Polynomial speedup for brute-force search, collision finding, graph problems (connectivity, shortest paths, minimum spanning trees, bipartiteness, network flows, finding subgraphs, etc.), algebra (associativity, commutativity, etc.), property testing, ...

Also cryptanalysis: Decoding random linear codes [Bernstein 10; Kachigar, Tillich 17], shortest vector problem [Laarhoven, Mosca, van de Pol 13], subset sum [Bernstein et al. 13], AES [Grassl et al. 16], bitcoin proof-of-work [Aggarwal et al. 17; Tessler, Byrnes 17]

Optimization

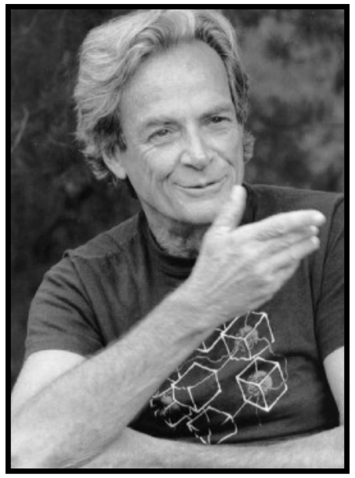
Quantum adiabatic optimization is a class of procedures for solving optimization problems by slowly changing the Hamiltonian to remain in its ground state [Farhi, Goldstone Gutmann, Sipser 00]

Successes:
- Quadratic speedup for unstructured search (with careful schedule)
- Can efficiently minimize some simple cost functions
- By tunneling through energy barriers, can succeed in some cases where simulating annealing fails

However:
- Can fail to efficiently minimize some cost functions by getting trapped in local minima
- Can sometimes be simulated classically (e.g., by quantum Monte Carlo)
- Overall, the power of this approach is far from clear

Related approach: "quantum approximate optimization algorithm". Discrete alternation between initial and final Hamiltonians can sometimes produce good approximate solutions quickly. May be promising, but the power of this approach is also unclear.

Quantum simulation



"... 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
Simulating physics with computers (1981)

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)

Applications: simulating chemical reactions (e.g., nitrogen fixation), properties of materials (e.g., high-T_c superconductivity), condensed matter physics, particle physics; also a tool for implementing other quantum algorithms

Long sequence of work led to optimal algorithm for simulating sparse Hamiltonains using *quantum signal processing* [Low, Chuang 16]

Linear algebra in Hilbert space

Basic computational problem: Solve for x in

$$A x = b$$

[Harrow, Hassidim, Lloyd 09]: Quantum algorithm running in time logarithmic in the size of A, provided

- A is given by a sparse Hamiltonian oracle and is well-conditioned
- b can be prepared as a quantum state
- it suffices to give the output x as a quantum state

Core of this algorithm: Quantum simulation

[Ambainis 10]: Improve dependence on condition number from quadratic to linear

[Childs, Kothari, Somma 15]: Improve dependence on precision from polynomial to logarithmic

Applications of quantum linear algebra

Solving differential equations

- [Berry 10]: Ordinary linear differential equations
- [Clader, Jacobs, Sprouse 13]: Preconditioned finite element method for PDEs (electromagnetic scattering)
- [Berry, Childs, Ostrander, Wang 17]: ODEs with poly(log) dependence on precision

Computing effective resistances

- [Wang 13]: Approximating effective resistances in sparse electrical networks with good expansion

Data analysis/machine learning

- [Wiebe, Braun, Lloyd 12]: Data fitting
- [Lloyd, Mohseni, Rebentrost 13]: Clustering
- [Rebentrost, Mohseni, Lloyd 13]: Support vector machines
- [Lloyd, Garnerone, Zanardi 14]: Computing Betti numbers

Convex optimization

- [Brandão, Svore 16; Apeldoorn, Gilyén, Gribling, de Wolf 17]: Quantum algorithms for linear and semidefinite optimization
- [Brandão, Kalev, Li, Lin, Svore, Wu 17]: Exponential speedup for low-rank constraints