From quantum simulation to quantum algorithms for linear algebra

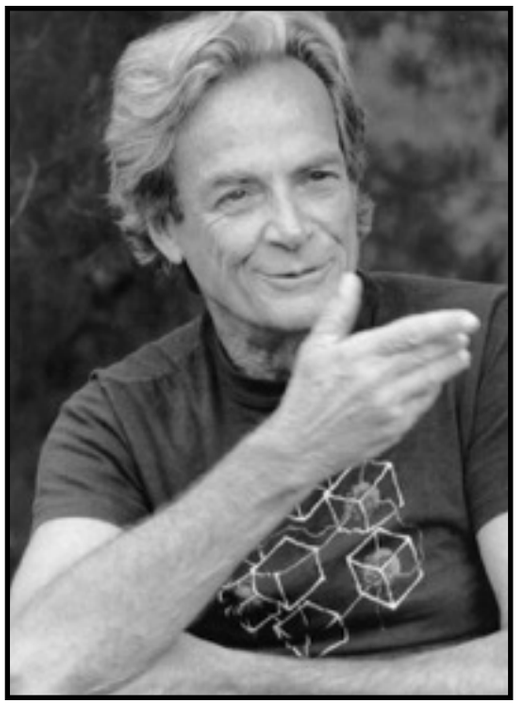
Andrew Childs

CS, UMIACS, & QuICS
University of Maryland

JOINT CENTER FOR QUANTUM INFORMATION AND COMPUTER SCIENCE

Based on joint work with Dominic Berry (Macquarie), Richard Cleve (Waterloo), Robin Kothari (MIT), and Rolando Somma (Los Alamos)

arXiv:1312.1414 / STOC 2014 arXiv:1412.4687 / to appear in PRL arXiv:1501.01715



"... 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)

Why simulate quantum mechanics?

Computational chemistry/physics

- chemical reactions
- properties of materials

Implementing quantum algorithms

- continuous-time quantum walk (e.g., for formula evaluation)
- adiabatic quantum computation (e.g., for optimization)
- linear/differential equations

Quantum dynamics

The dynamics of a quantum system are determined by its *Hamiltonian*.

$$i\frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle = H|\psi(t)\rangle$$
$$\Downarrow$$
$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$

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 by performing measurements on the state, it can answer questions that (apparently) a classical computer cannot

Local and sparse Hamiltonians

Local Hamiltonians [Lloyd 96]

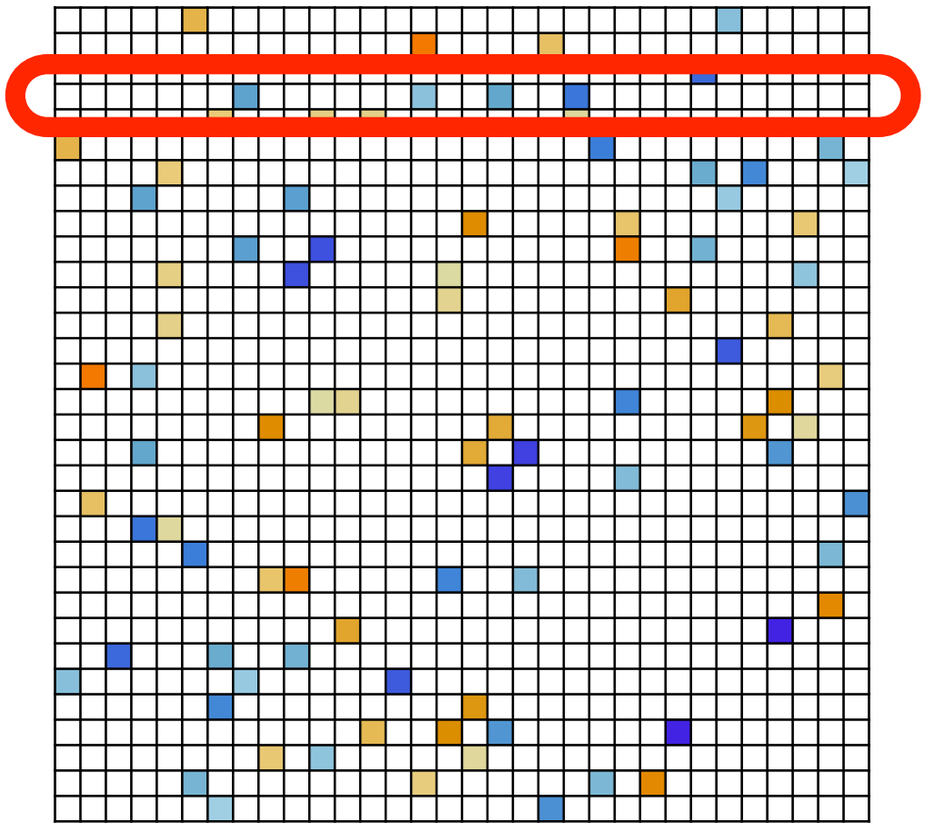
$H = \sum_{j=1}^{m} H_j$ where each H_j acts on $k = O(1)$ qubits

Sparse Hamiltonians [Aharonov, Ta-Shma 03]

At most d nonzero entries per row, $d = \mathrm{poly}(\log N)$ (where H is $N \times N$)

In any given row, the location of the jth nonzero entry and its value can be computed efficiently (or is given by a black box)

$H =$



Note: A k-local Hamiltonian with m terms is d-sparse with $d = 2^k m$

Previous simulation methods

Product formulas

- Decompose Hamiltonian into a sum of terms that are easy to simulate
- Recombine the terms by alternating between them

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

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

$$\vdots$$

Quantum walk

- Define an easy-to-implement unitary operation (a step of a quantum walk) whose spectrum is related to the Hamiltonian
- Use phase estimation to obtain information about the spectrum
- Introduce phases to give the desired evolution

Complexity of both approaches: $\mathrm{poly}(t, \log N, d, 1/\epsilon)$

Linear combinations of unitaries

LCU Lemma: Given the ability to perform unitaries V_j with unit complexity, one can perform the operation $U = \sum_j \beta_j V_j$ with complexity $O(\sum_j |\beta_j|)$. Furthermore, if U is (nearly) unitary then this implementation can be made (nearly) deterministic.

Main ideas:

- Using controlled-V_j operations, implement U with some amplitude:

$$|0\rangle|\psi\rangle \mapsto \sin\theta|0\rangle U|\psi\rangle + \cos\theta|\Phi\rangle$$

- Boost the amplitude for success by oblivious amplitude amplification

Quantum walk simulation

Each eigenvalue λ of H corresponds to two eigenvalues $\pm e^{\pm i \arcsin \lambda}$ of an easily-implemented quantum walk operator (with eigenvectors also simply related to those of H)

Strategy: Use phase estimation to determine and correct the phase

Complexity: $O(\tau/\sqrt{\epsilon})$ $\tau := d\|H\|_{\max} t$

[Childs 10], [Berry, Childs 12]

Linear combination of quantum walk steps

Another approach: find coefficients so that

$$e^{-iH} \approx \sum_{k=-K}^{K} \beta_k U^k \qquad \text{(within an appropriate subspace)}$$

and implement this using the LCU Lemma

By a generating series for Bessel functions,

$$e^{-i\lambda t} = \sum_{k=-\infty}^{\infty} J_k(-t)\, e^{ik \arcsin \lambda}$$

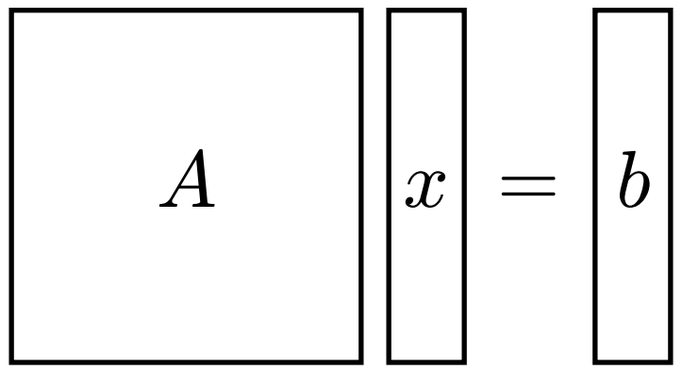
Coefficients drop off rapidly for large k, so we can truncate the series

Query complexity of this approach: $O\left(\tau \frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\right)$

$$\tau := d\|H\|_{\max} t$$

Quantum algorithms for linear algebra

Basic computational problem: Solve for x in



[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 is available as a quantum state
- it suffices to give the output x as a quantum state

Core of this algorithm: Quantum simulation

Question: Can we further improve algorithms for linear systems using recent advances in quantum simulation?

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)

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

Recent survey by Aaronson: scottaaronson.com/papers/qml.pdf

Computational power of q. linear algebra

Does the quantum algorithm for linear systems give an exponential speedup over classical computation?

[HHL 09]: The problem solved by the quantum linear systems algorithm (sparse, implicitly-specified, well-conditioned A; input quantum state b; output quantum state x) is BQP-complete

What can we say about the complexity of some of its applications?

Example: Is the problem of approximately computing effective resistances in a sparse graph with good expansion

- BQP-complete?
- Efficiently solvable by a classical algorithm?

Outlook

Improved simulation algorithms

- Optimal tradeoff for sparse Hamiltonian simulation
- Faster algorithms for structured problems
- Simulating open quantum systems

Applications to simulating physics

- What is the cost in practice for simulating molecular systems?
- How do recent algorithms compare to naive methods?

New quantum algorithms

- Improved algorithms for linear systems
- Computational power of applications of linear systems
- New applications of linear systems
- Other quantum algorithms from quantum simulation