Algorithmic advances in quantum simulation

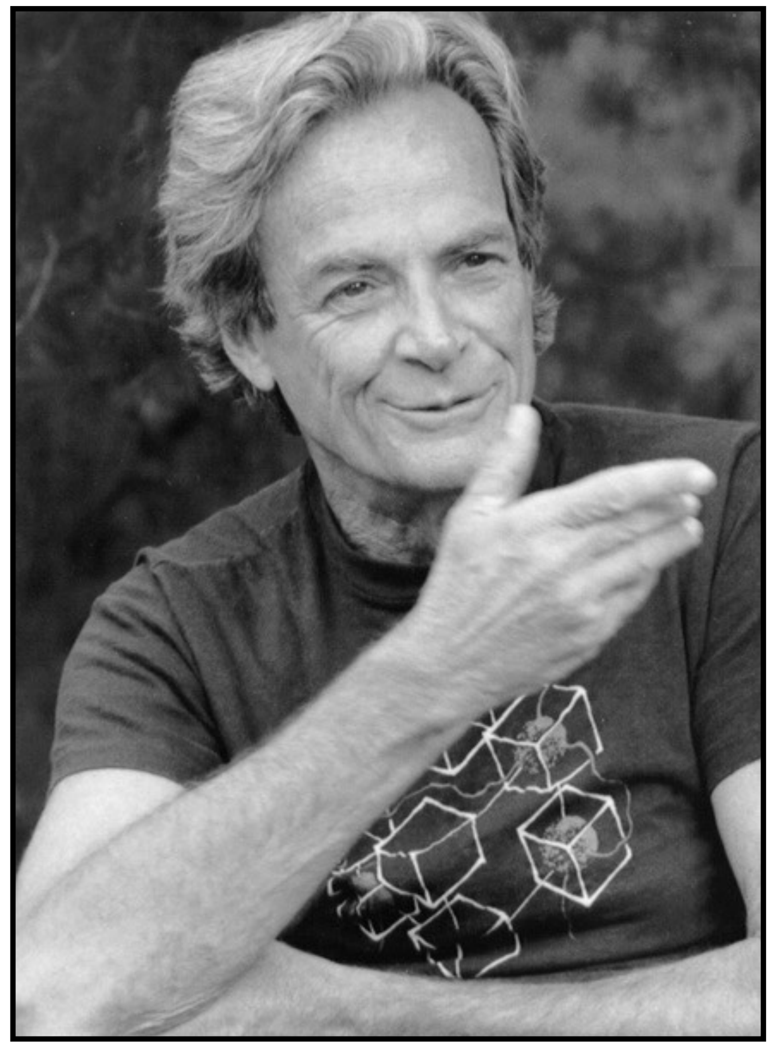
Andrew Childs
University of Maryland

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

UMIACS
University of Maryland
Institute for Advanced
Computer Studies

JOINT CENTER FOR
QUANTUM INFORMATION
AND COMPUTER SCIENCE

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 (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)

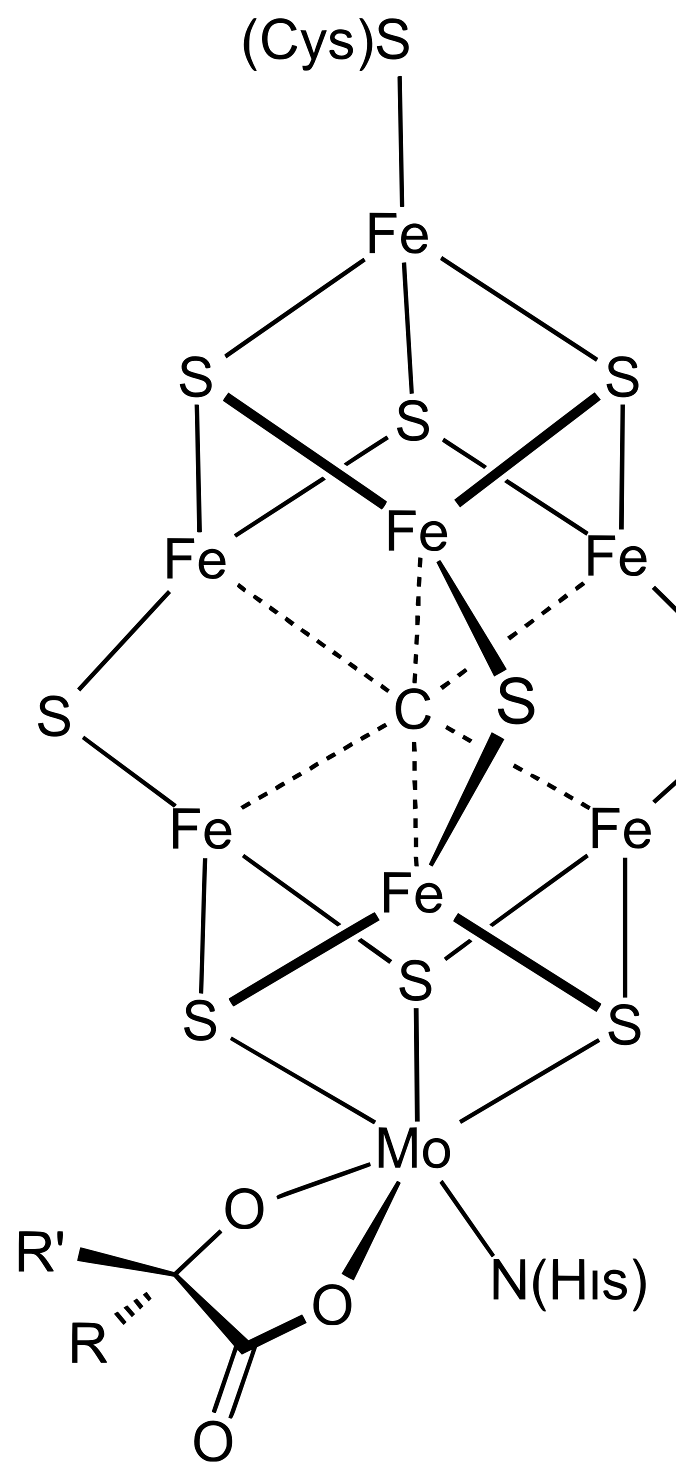
A classical computer cannot even represent the state efficiently.

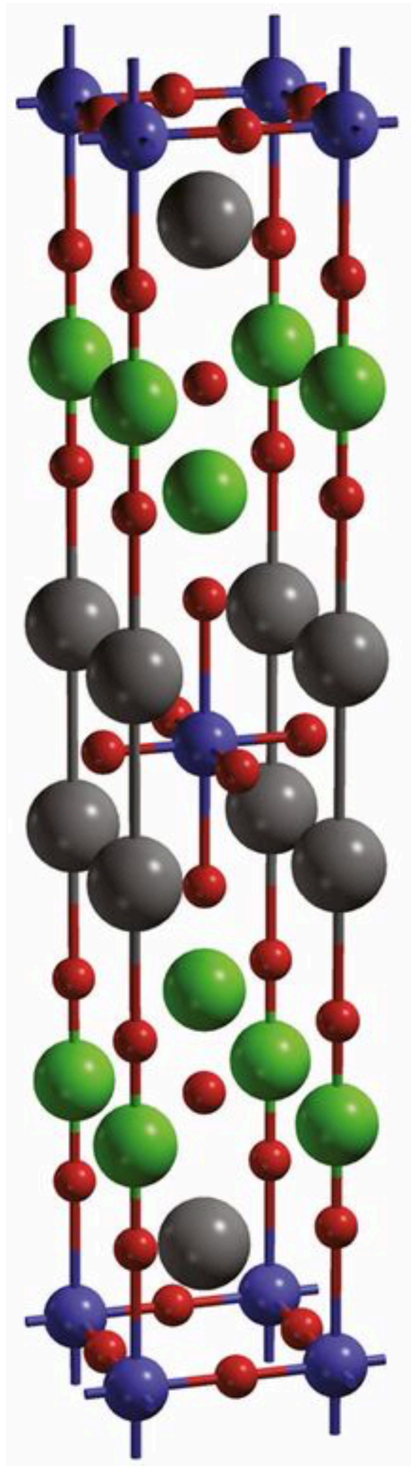
A quantum computer cannot produce a complete description of the state.

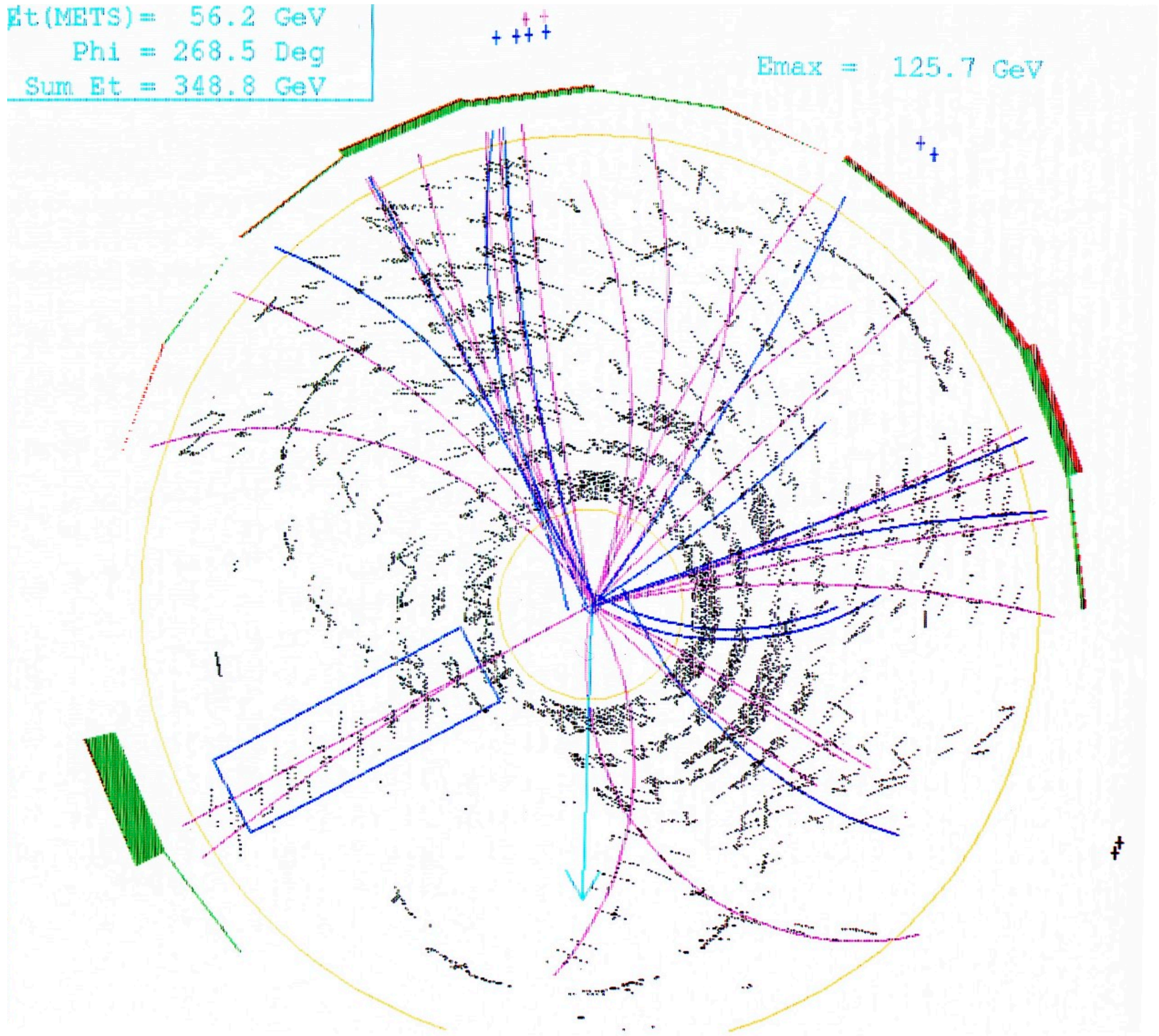
But given succinct descriptions of
- the initial state (suitable for a quantum computer to prepare it efficiently) and
- a final measurement (say, measurements of the individual qubits in some basis),

a quantum computer can efficiently answer questions that (apparently) a classical one cannot.

Computational quantum physics





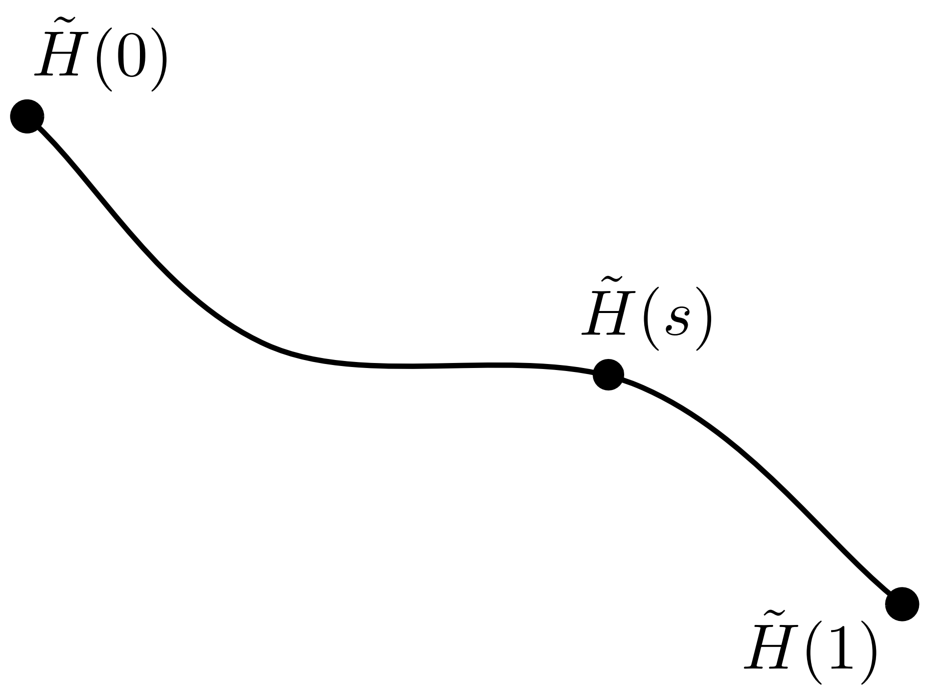


chemical reactions
(e.g., nitrogen fixation)

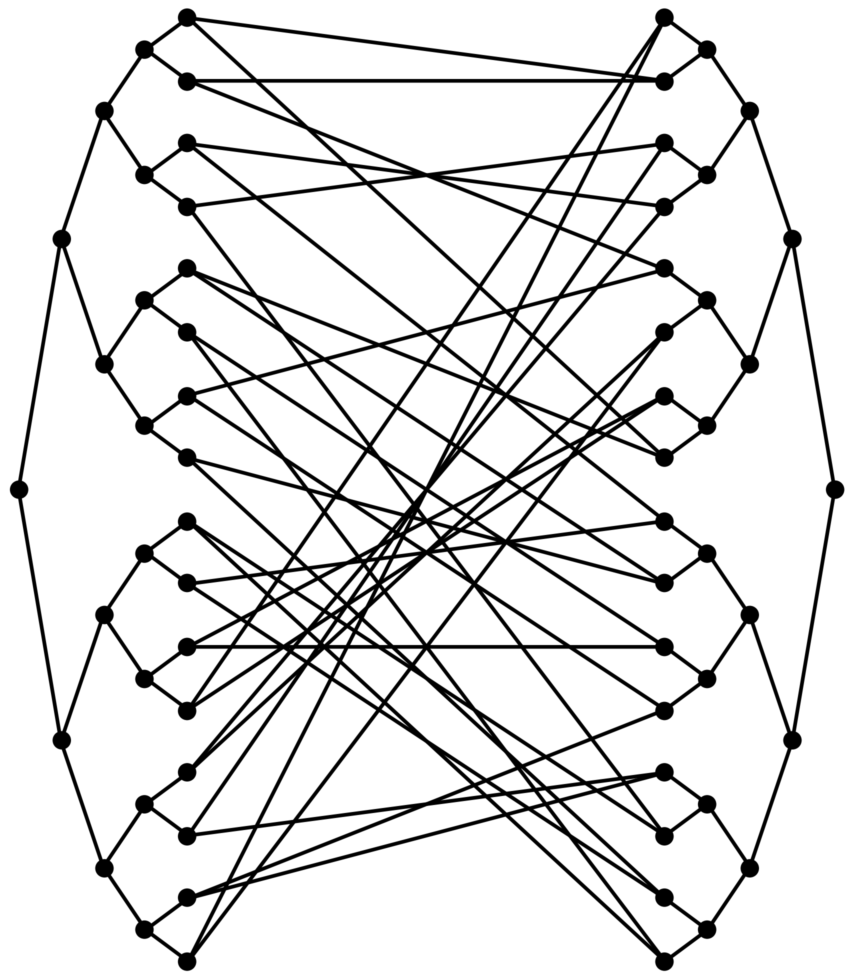
condensed matter physics/
properties of materials

nuclear/particle
physics

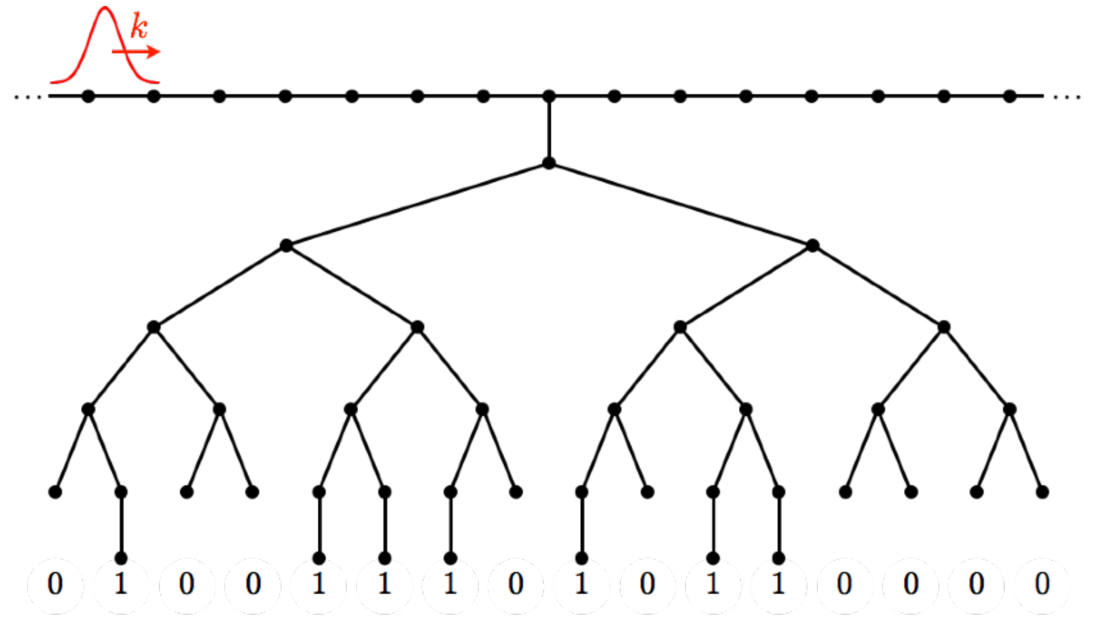
Implementing quantum algorithms



adiabatic optimization



exponential speedup by quantum walk



evaluating Boolean formulas

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

linear/ differential equations, convex optimization

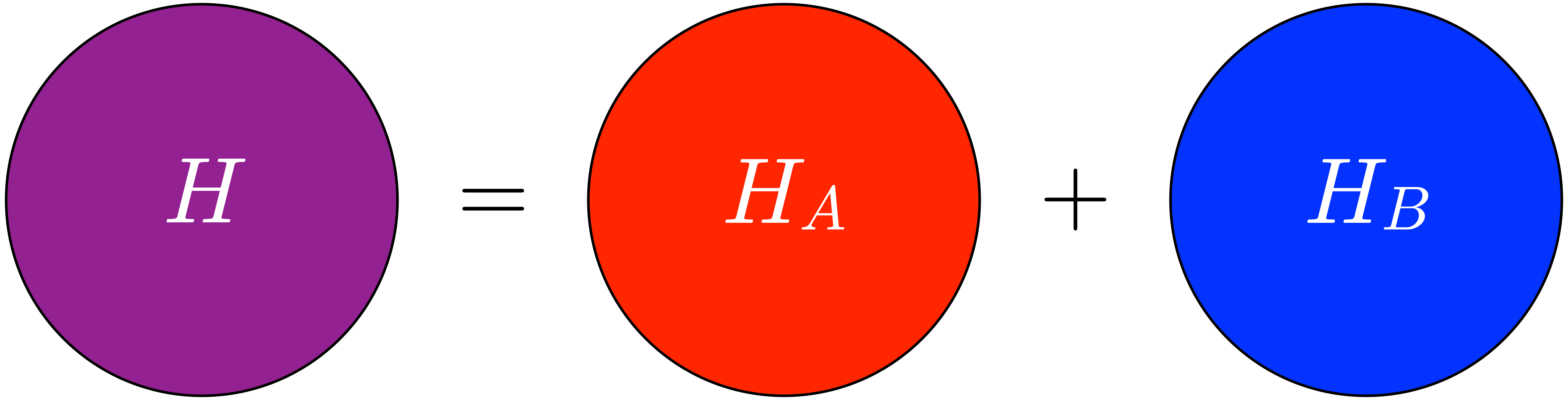
Product formula simulation

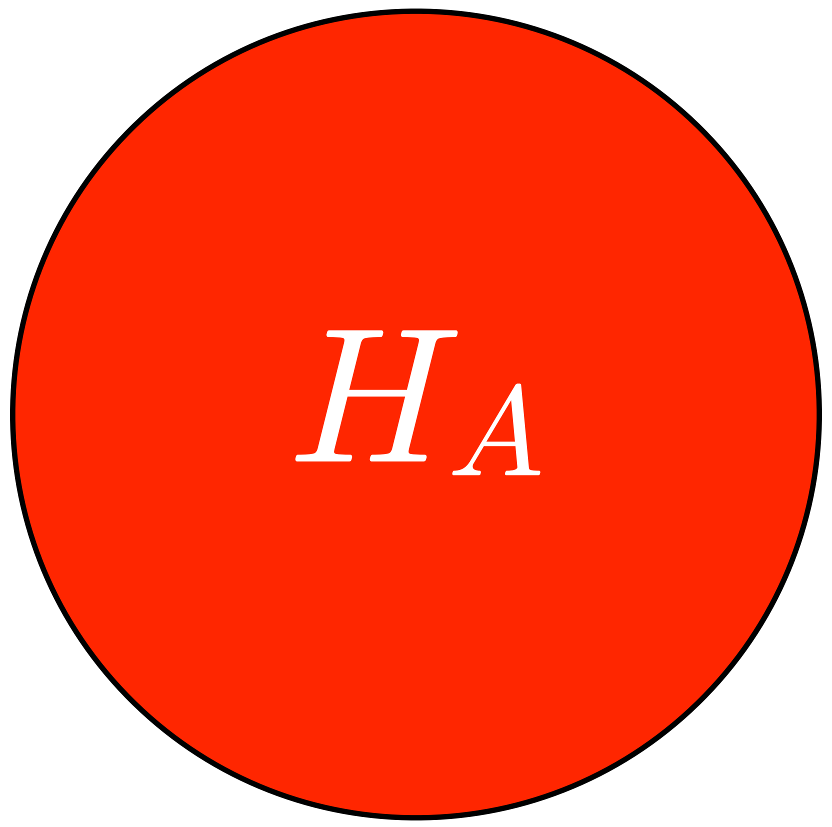
Suppose we want to simulate $H = \sum_{\ell=1}^{L} H_\ell$

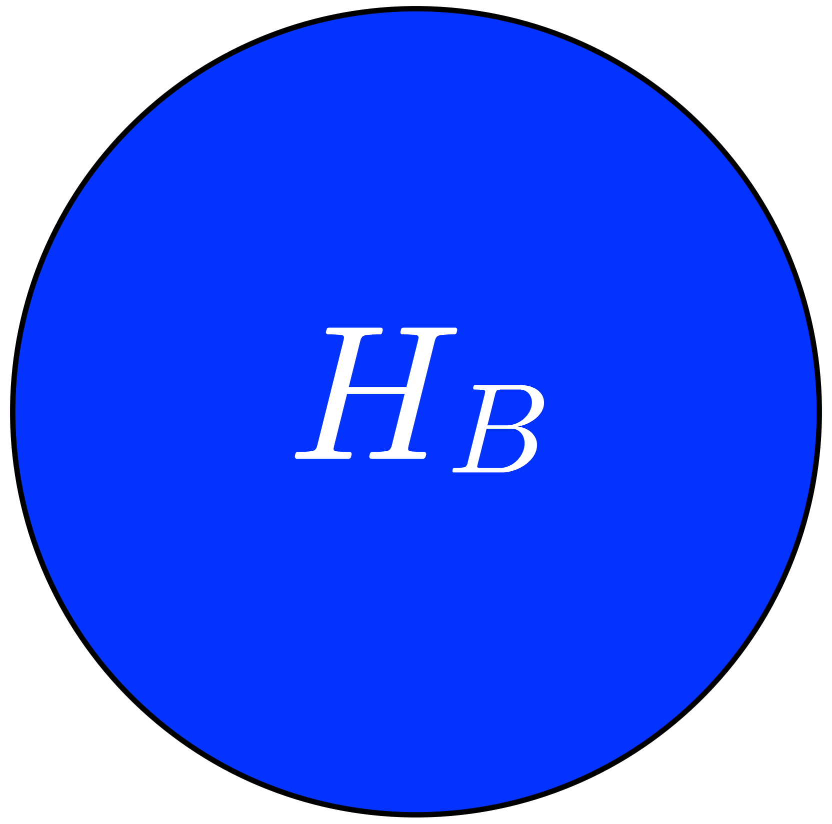
Combine individual simulations with the Lie product formula. E.g., with two terms:

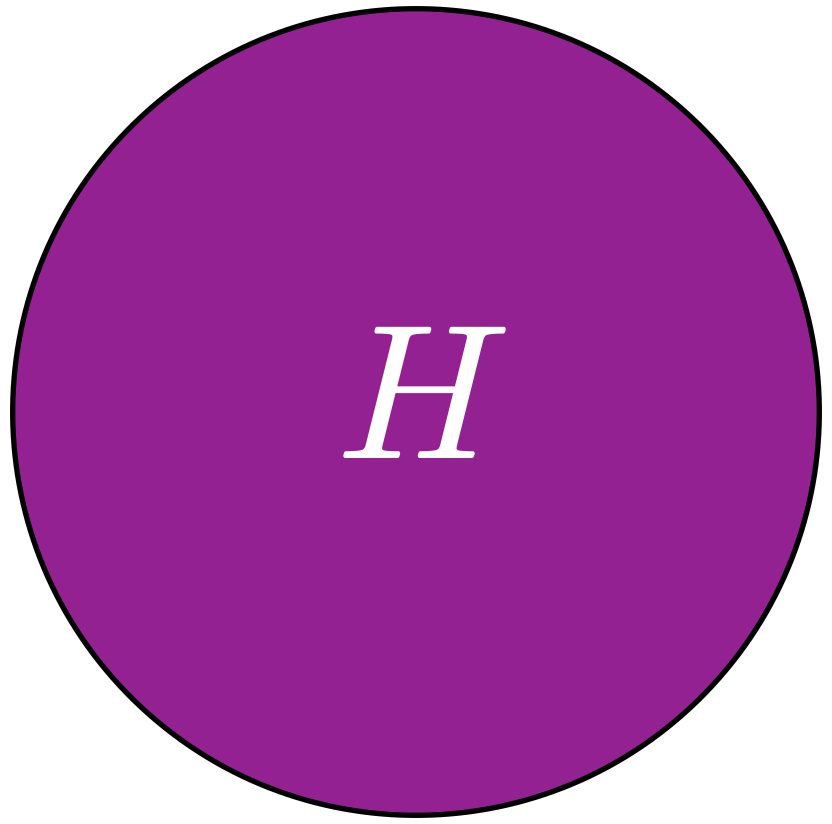
$$\lim_{r\to\infty} \left(e^{-iAt/r} e^{-iBt/r}\right)^r = e^{-i(A+B)t}$$

[Lloyd 96]


H
=
H_A
+
H_B


H_A


H_B


H

Product formula simulation

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 $r = O\big((\|H\|t)^2/\epsilon\big)$

[Lloyd 96]

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

E.g., second order:

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

Systematic expansions to arbitrary order are known [Suzuki 92]

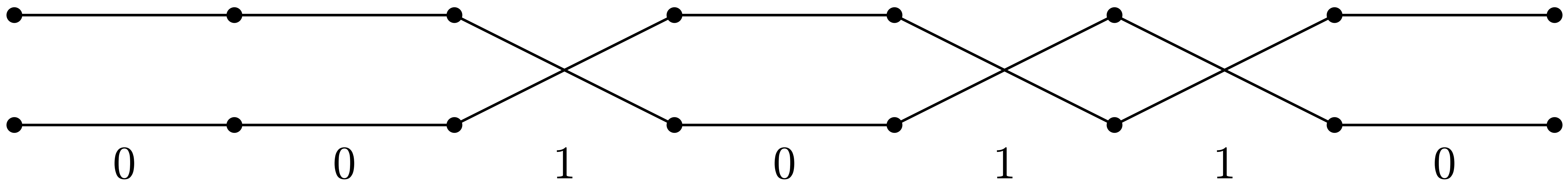
Using the $2k$th order expansion, the number of exponentials required for an approximation with error at most ϵ is at most

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

[Berry, Ahokas, Cleve, Sanders 07]

Simulating quantum mechanics in real time

No fast-forwarding theorem: Simulating Hamiltonian dynamics for time t requires $\Omega(t)$ gates.



[Berry, Ahokas, Cleve, Sanders 05]

Complexity of kth order product formula simulation is $O(5^{2k}t^{1+1/2k})$.

Can we give an algorithm with complexity precisely $O(t)$?

Pro: Systems simulate their own dynamics in real time!
Con: Mismatch between continuous-time dynamics and the discrete-time circuit model.

Hamiltonian simulation by quantum walk

Quantum walk corresponding to H

Alternately reflect about $\mathrm{span}\{|\psi_j\rangle\}_{j=1}^N$,

$$|\psi_j\rangle := |j\rangle \otimes \left(\nu \sum_{k=1}^{N} \sqrt{H_{jk}^*}|k\rangle + \nu_j|N+1\rangle\right),$$

and swap the two registers.

If H is sparse, this walk is easy to implement.

Spectral theorem: Each eigenvalue λ of H corresponds to two eigenvalues $\pm e^{\pm i \arcsin \lambda}$ of the walk operator (with eigenvectors closely related to those of H).

Simulation by phase estimation

$$\begin{aligned} |\lambda\rangle &\mapsto |\lambda\rangle|\widetilde{\arcsin \lambda}\rangle && \text{(phase estimation)} \\ &\mapsto e^{-i\lambda t}|\lambda\rangle|\widetilde{\arcsin \lambda}\rangle \\ &\mapsto e^{-i\lambda t}|\lambda\rangle && \text{(inverse phase est)} \end{aligned}$$

Theorem: $O(t/\sqrt{\epsilon})$ steps of this walk suffice to simulate H for time t with error at most ϵ.

[Childs10; Berry, Childs 12]

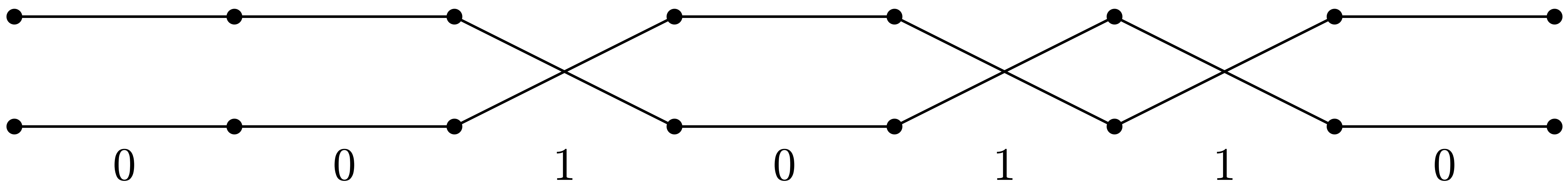
High-precision simulation?

Can we improve the dependence on ϵ?

Many approximate computations can be done with complexity $\mathrm{poly}(\log(1/\epsilon))$:

- computing numerical constants (e.g., π)
- boosting a bounded-error subroutine
- Solovay-Kitaev circuit synthesis
- and more…

Lower bound (based on the *unbounded-error* query complexity of parity): $\Omega\left(\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right)$



Quantum walk simulation: $O(1/\sqrt{\epsilon})$

Product formulas ($2k$th order): $O(5^{2k}\epsilon^{-2k})$

Can we do better?

Hamiltonian simulation by linear combinations of unitaries

Main idea: Directly implement the series

$$e^{-iHt} = \sum_{k=0}^{\infty} \frac{(-iHt)^k}{k!}$$
$$\approx \sum_{k=0}^{K} \frac{(-iHt)^k}{k!}$$

Write $H = \sum_\ell \alpha_\ell H_\ell$ with H_ℓ unitary.

Then

$$\sum_{k=0}^{K} \sum_{\ell_1,\ldots,\ell_k} \frac{(-it)^k}{k!} \alpha_{\ell_1} \cdots \alpha_{\ell_k} H_{\ell_1} \cdots H_{\ell_k}$$

is a linear combination 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:

- 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*

Query complexity: $O\left(t\frac{\log(t/\epsilon)}{\log\log(t/\epsilon)}\right)$

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

Tradeoff between t and ϵ

Combining known lower bounds on the complexity of simulation as a function of t and ϵ gives

$$\Omega\Big(t + \frac{\log \frac{1}{\epsilon}}{\log\log \frac{1}{\epsilon}}\Big) \quad \text{vs. upper bound of} \quad O\Big(t \frac{\log \frac{t}{\epsilon}}{\log\log \frac{t}{\epsilon}}\Big)$$

An alternative method for implementing a linear combination of unitary operations, *quantum signal processing*, gives an optimal tradeoff. [Low, Chuang 16, 17]

Main idea: Encode the eigenvalues of H in a two-dimensional subspace; use a carefully-chosen sequence of single-qubit rotations to manipulate those eigenvalues.

Computing the rotation angles is challenging, but can be done efficiently (classically) [Haah 18]. Recent approaches are faster [Dong, Meng, Whaley, Lin 20; Chao, Ding, Gilyén, Huang, Szegedy 20].

Quantum signal processing (and more general *quantum singular value transformation*) gives a powerful framework for designing other quantum algorithms [Gilyén, Su, Low, Wiebe 19].

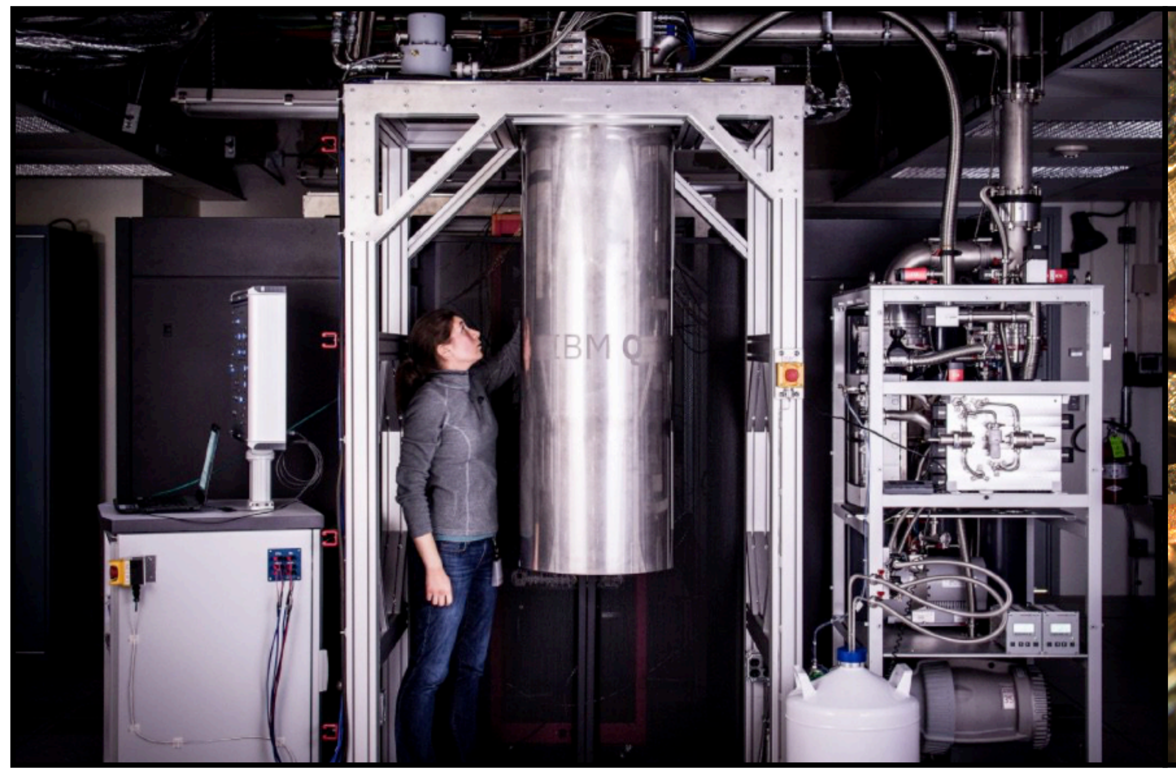
Algorithm comparison

t: evolution time
ϵ: allowed error
d: sparsity

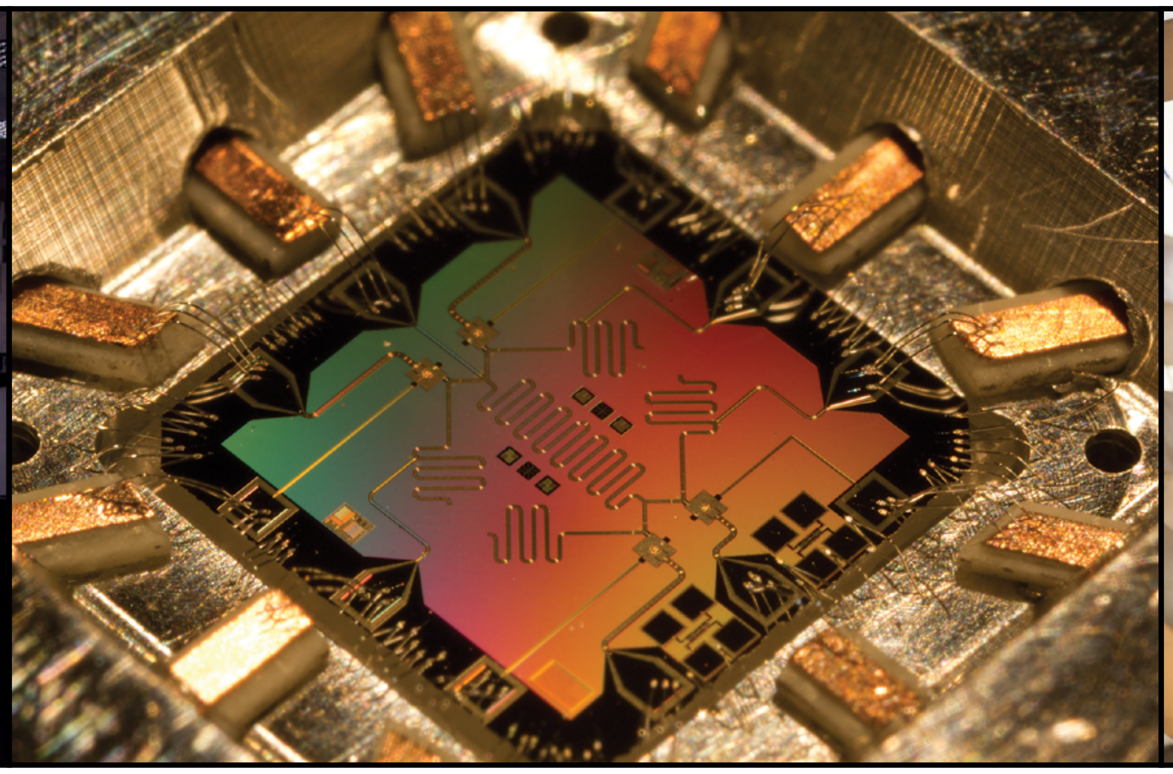
Algorithm	Query complexity	Gate complexity
Product formula, 1st order	$O(d^4t^2/\epsilon)$	$O(d^4t^2/\epsilon)$
Product formula, (2k)th order	$O\left(5^{2k}d^3t(\frac{dt}{\epsilon})^{1/2k}\right)$	$O\left(5^{2k}d^3t(\frac{dt}{\epsilon})^{1/2k}\right)$
Quantum walk	$O(dt/\sqrt{\epsilon})$	$O(dt/\sqrt{\epsilon})$
Fractional-query simulation	$O\left(d^2t\frac{\log(dt/\epsilon)}{\log\log(dt/\epsilon)}\right)$	$O\left(d^2t\frac{\log^2(dt/\epsilon)}{\log\log(dt/\epsilon)}\right)$
Taylor series	$O\left(d^2t\frac{\log(dt/\epsilon)}{\log\log(dt/\epsilon)}\right)$	$O\left(d^2t\frac{\log^2(dt/\epsilon)}{\log\log(dt/\epsilon)}\right)$
Linear combination of q. walk steps	$O\left(dt\frac{\log(dt/\epsilon)}{\log\log(dt/\epsilon)}\right)$	$O\left(dt\frac{\log^{3.5}(dt/\epsilon)}{\log\log(dt/\epsilon)}\right)$
Quantum signal processing	$O\left(dt+\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right)$	$O\left(dt+\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right)$

OPTIMAL!

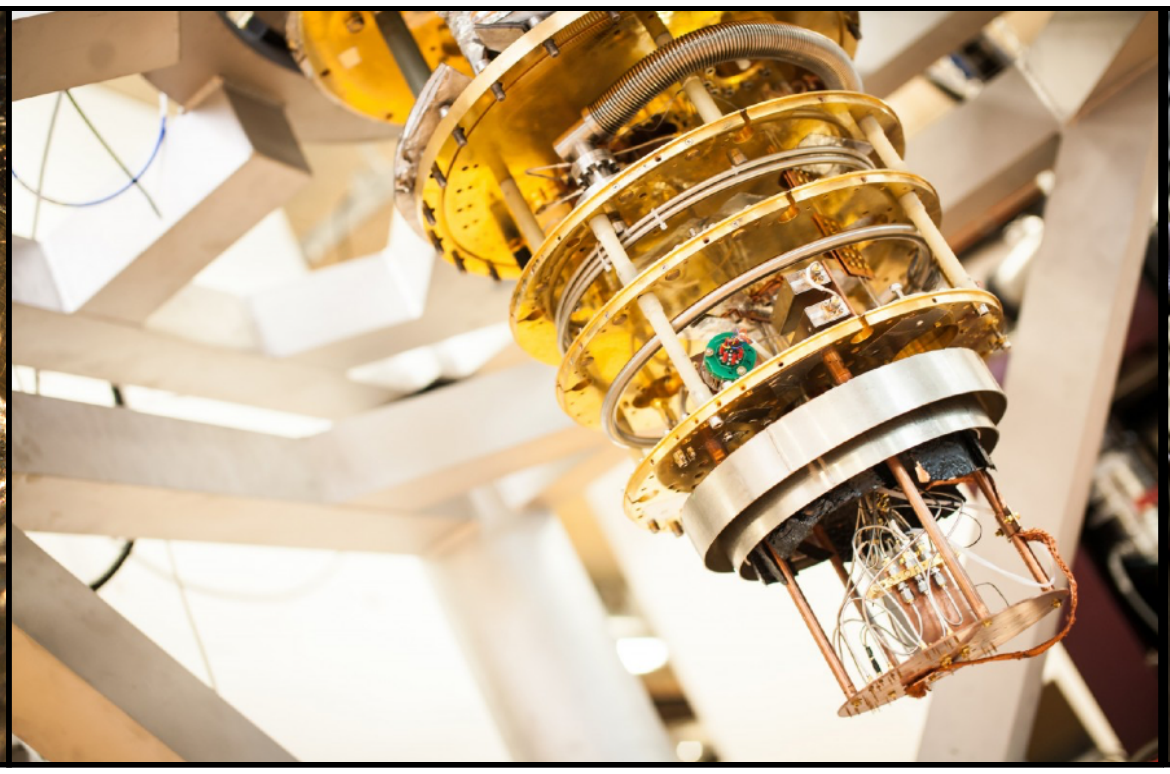
Toward practical quantum speedup



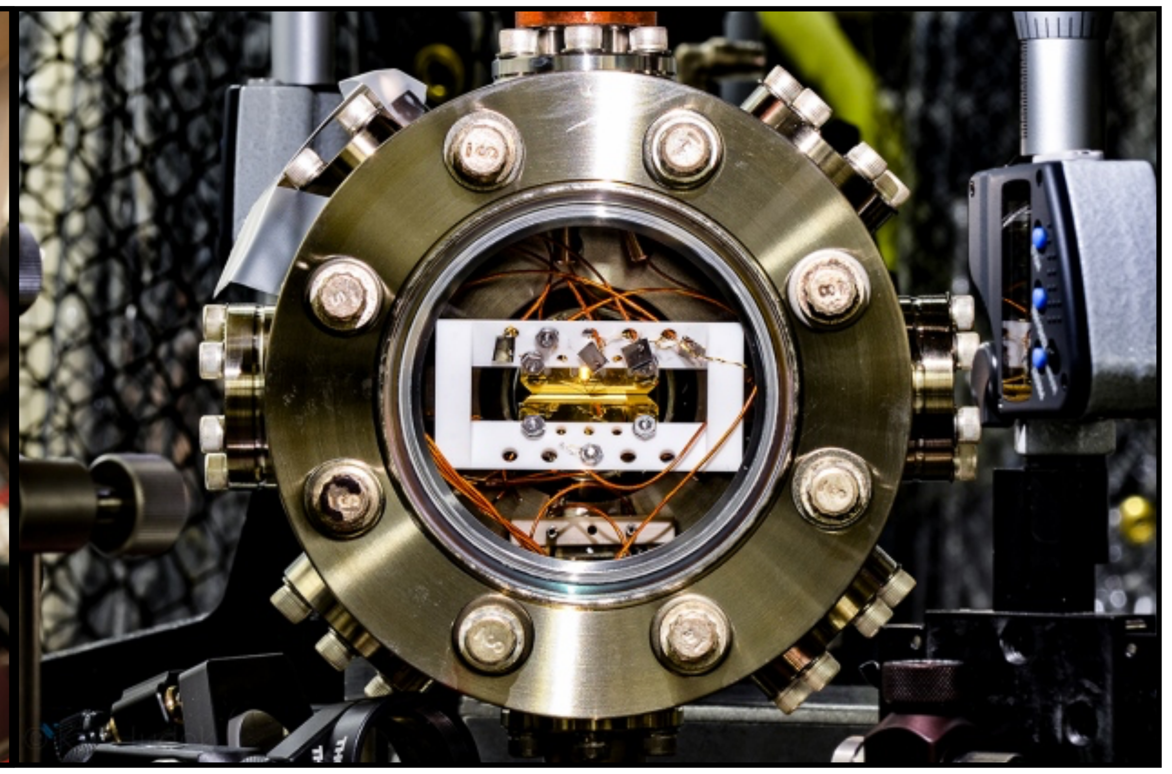
IBM



Google



Delft



Maryland

Important early goal: demonstrate quantum computational advantage

… but can we find a *practical* application of near-term devices?

Challenges

- Improve experimental systems
- Improve algorithms and their implementation, making the best use of available hardware

Goal: Produce concrete resource estimates for the simplest possible practical application of quantum computers

What to simulate?

~~Quantum chemistry?~~ Spin systems!

Heisenberg model on a ring: $H = \sum_{j=1}^{n} (\vec{\sigma}_j \cdot \vec{\sigma}_{j+1} + h_j \sigma_j^z)$ $h_j \in [-h, h]$ uniformly random

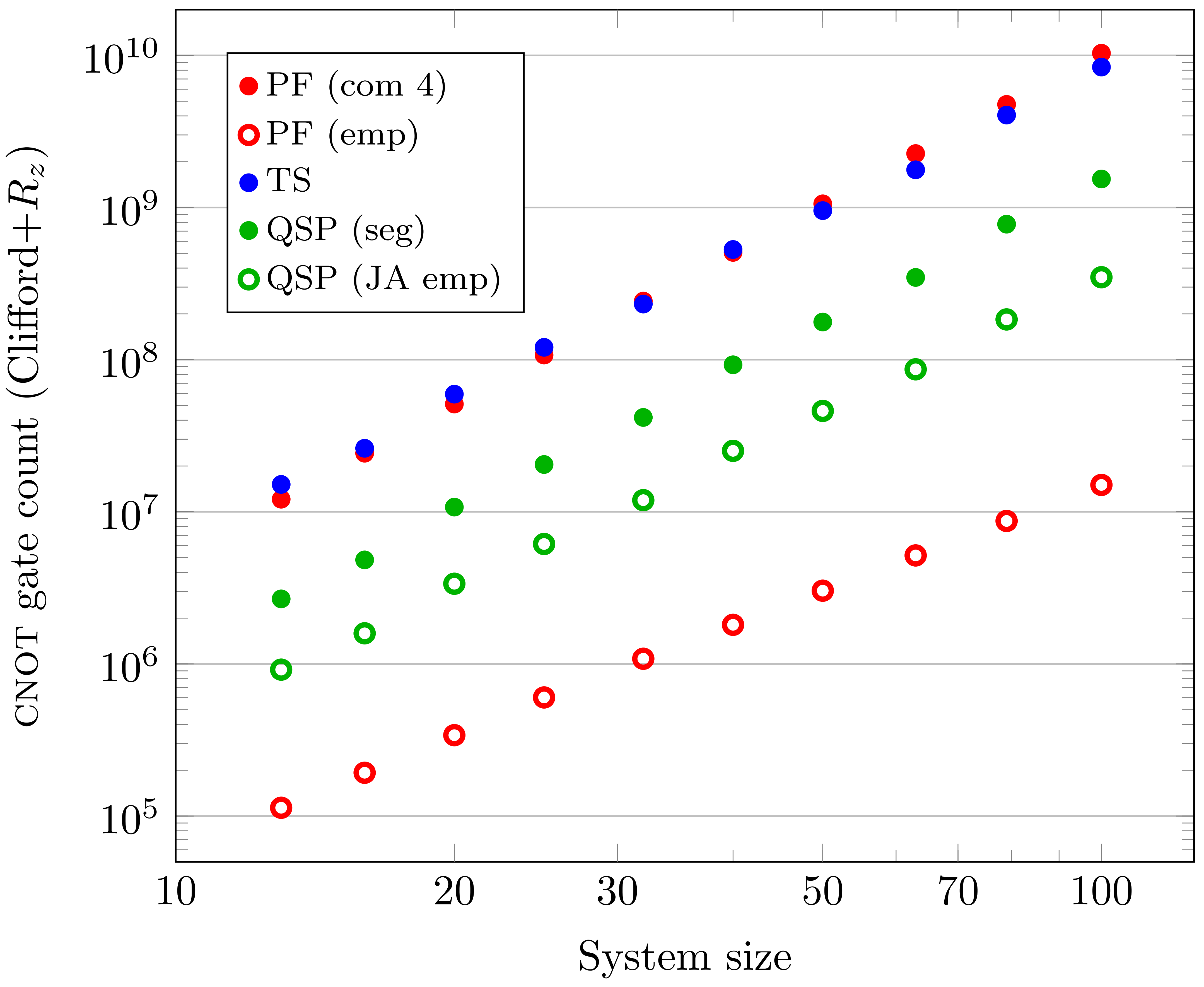
This provides a model of *self-thermalization* and *many-body localization*.

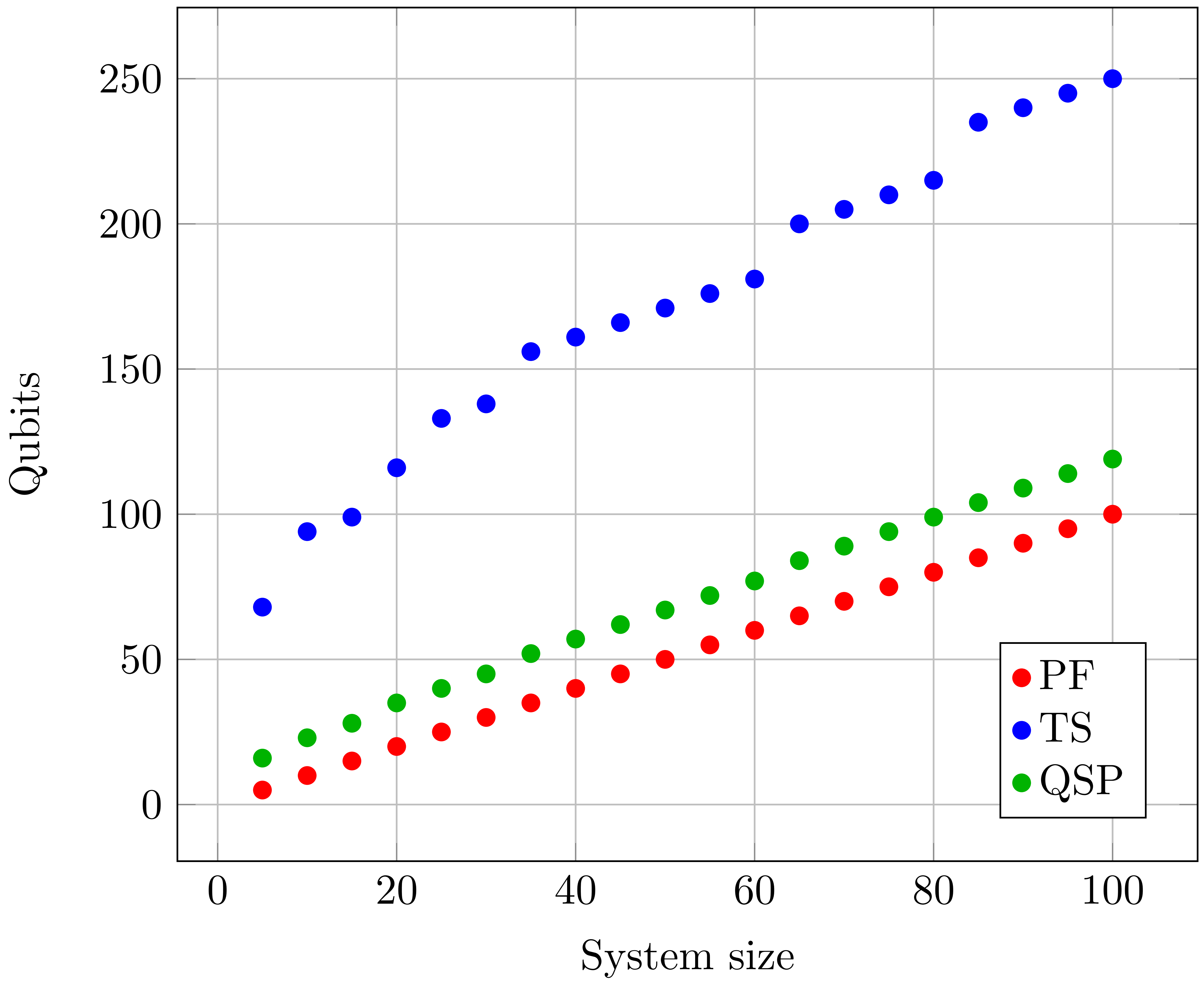
The transition between thermalized and localized phases (as a function of h) is poorly understood. Most extensive numerical study: fewer than 25 spins. [Luitz, Laflorencie, Alet 15]

Could explore the transition by preparing a simple initial state, evolving, and performing a simple final measurement. Focus on the cost of simulating dynamics.

For concreteness: $h = 1, \quad t = n, \quad \epsilon = 10^{-3}, \quad 20 \leq n \leq 100$

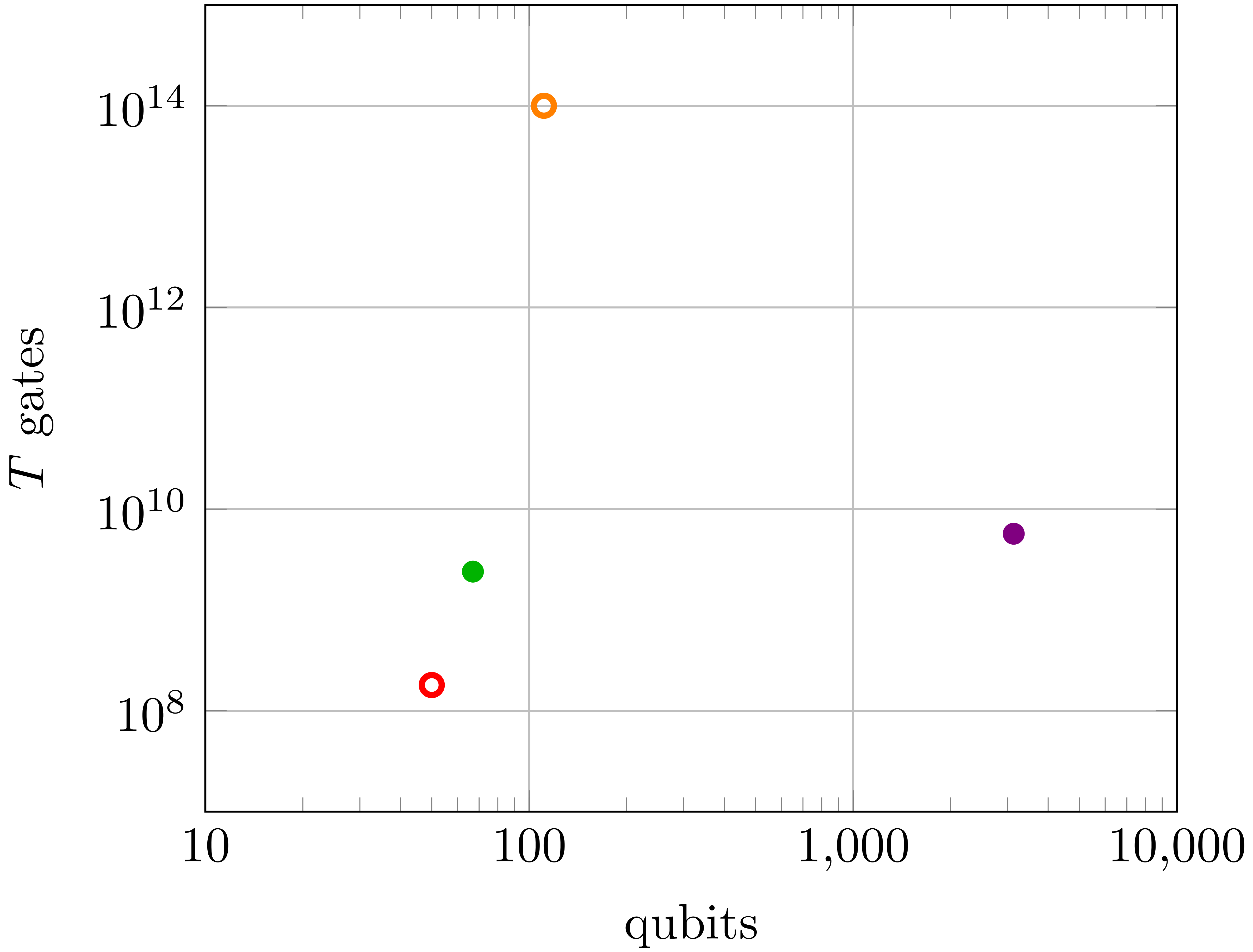
Resource estimates





[Childs, Maslov, Nam, Ross, Su 18]

Comparison



Factoring a 1024-bit number [Kutin 06]

- 3132 qubits
- 5.7×10^9 T gates

Simulating FeMoco [Reiher et al. 16]

- 111 qubits
- 1.0×10^{14} T gates

Simulating 50 spins (segmented QSP)

- 67 qubits
- 2.4×10^9 T gates

Simulating 50 spins (PF6 empirical)

- 50 qubits
- 1.8×10^8 T gates

[Childs, Maslov, Nam, Ross, Su 18]

Lattice Hamiltonians

We've focused on the complexity as a function of t (evolution time) and ϵ (precision).
What about the dependence on system size?

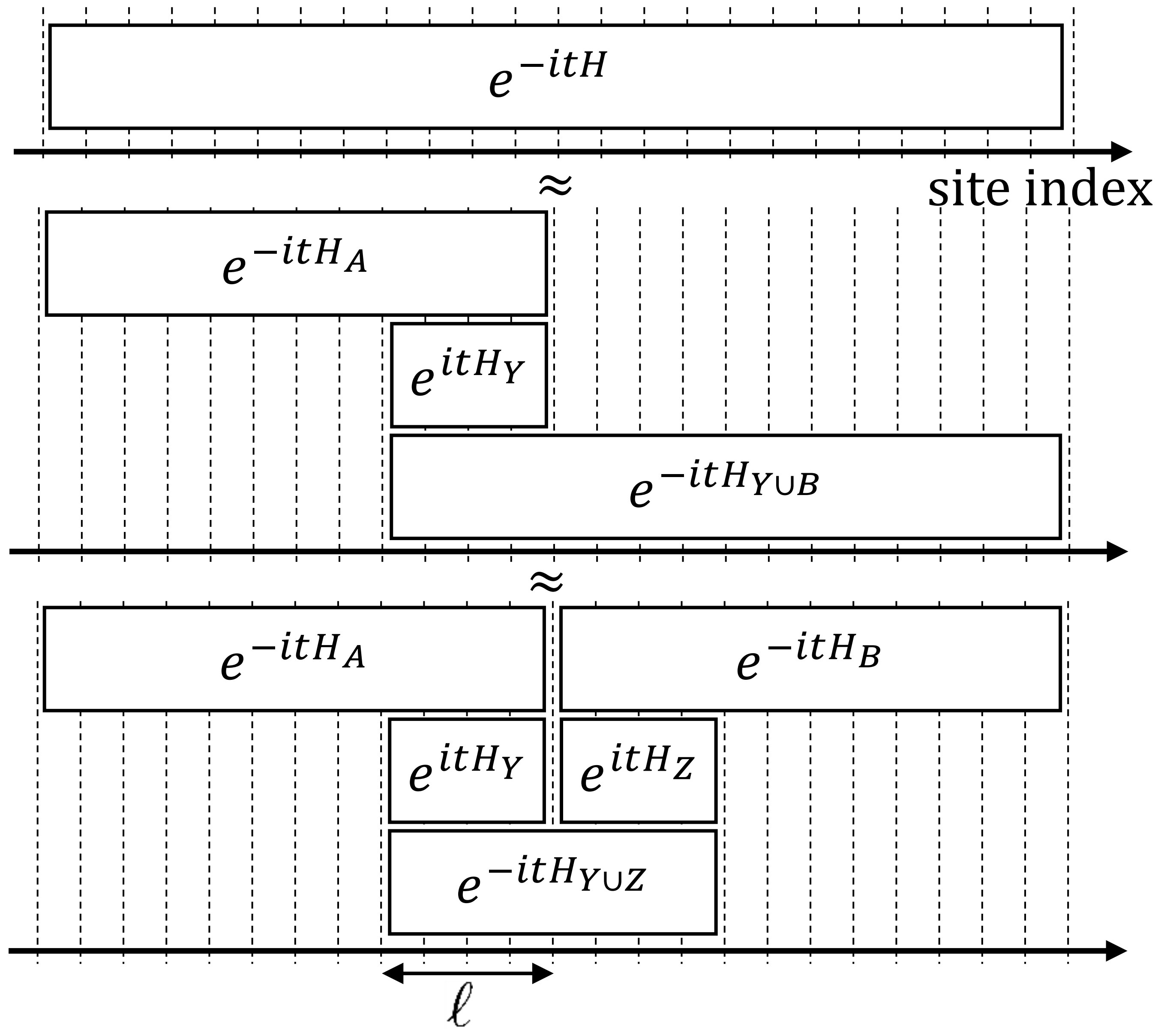
Consider a system of n spins with nearest-neighbor interactions on a grid of fixed dimension. To simulate for constant time, best previous methods (TS, QSP, high-order PF) give:

- total number of gates: $O(n^2)$
- circuit depth (execution time with parallel gates): $O(n)$

Execution time should not have to be extensive!

Recent improvement: simulation with $\tilde{O}(n)$ gates, $\tilde{O}(1)$ depth (optimal!) [Haah, Hastings, Kothari, Low 18]

- Lieb-Robinson bound limits the speed of propagation
- Simulate small regions with negative-time evolutions to correct the boundaries



Local error analysis

In fact, product formulas achieve nearly the same complexity!

Main technique: *local error analysis* provides a convenient integral representation of the error

[Descombes, Thalhammer 10]

Example (first order):

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

For an n-site lattice system, letting A = even terms and B = odd terms, we find a simulation error of $O(nt^2)$, so $O(n^2t^2)$ gates suffice to simulate with constant accuracy (vs. $O(n^3t^2)$ with standard analysis).

Generalizations give similar (though more complicated!) expressions for the error in higher-order product formulas.

Complexity at order p: $O\big((nt)^{1+\frac{1}{p}}\big)$ (vs. $O\big(n(nt)^{1+\frac{1}{p}}\big)$ with standard analysis)

[Childs, Su 19]

A theory of Trotter error

Local error analysis can be generalized to give tight bounds on the error of product formula approximations depending on commutators of the terms.

Theorem. A pth order product formula approximates the evolution of $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$ with additive error $O(\alpha t^{p+1})$ where

$$\alpha := \sum_{\gamma_1,\ldots,\gamma_{p+1}} \left\| [H_{\gamma_{p+1}}, [\cdots, [H_{\gamma_2}, H_{\gamma_1}] \cdots]] \right\|.$$

Therefore $O(\Gamma \alpha^{1/p} t^{1+1/p})$ gates suffice to give a simulation with constant accuracy.

Applications:

- Tighter rigorous analysis of product formulation simulations (e.g., only off by factor of ~5 for 50-qubit Heisenberg model)
- Simpler simulation of plane-wave electronic structure, nearly matching interaction picture simulation
- Simulation of k-local Hamiltonians with better norm scaling than qubitization
- Faster simulation of power-law interactions
- Faster hybrid quantum/classical simulation of clustered Hamiltonians
- Tighter analysis of quantum Monte Carlo methods

[Childs, Su, Tran, Wiebe, Zhu 19; talk at https://ter.ps/trottertalk]

Randomized simulation

Another approach to speeding up simulation: introduce classical randomness

Example:

$$
\begin{aligned}
e^{-i(A+B)t} &= I - i(A+B)t - \tfrac{1}{2}(A^2 + AB + BA + B^2)t^2 + O(t^3) \\
e^{-iAt}e^{-iBt} &= I - i(A+B)t - \tfrac{1}{2}(A^2 + 2AB + B^2)t^2 + O(t^3) \\
e^{-iBt}e^{-iAt} &= I - i(A+B)t - \tfrac{1}{2}(A^2 + 2BA + B^2)t^2 + O(t^3)
\end{aligned}
$$

$$\Downarrow$$

$$\tfrac{1}{2}(e^{-iAt}e^{-iBt} + e^{-iBt}e^{-iAt}) = e^{-i(A+B)t} + O(t^3)$$

[Zhang 12]

Mixing lemma [Campbell 17, Hastings 17]: Error of the average operation is linear in the average error, quadratic in the error of individual operations.

Randomly permuting terms in a higher-order product formula also improves the approximation (though not the order of the formula). [Childs, Ostrander, Su 18]

It can also be advantageous to sample terms of the Hamiltonian nonuniformly. [Campbell 18]
Gives faster simulations of strongly time-dependent Hamiltonians. [Berry, Childs, Su, Wang, Wiebe 20]

Outlook

Develop applications of quantum simulation to physics/chemistry

- Quantum chemistry
- Condensed matter
- Nuclear/particle physics

Improve quantum algorithms for Hamiltonian simulation

- Tighter error bounds for product formulas (improve local error analysis; go beyond the triangle inequality)
- Faster simulation methods for structured Hamiltonians
- More efficient synthesis of the QSP circuit

Explore prospects for near-term implementations

- Resource estimates under realistic hardware constraints
- Can we perform classically hard simulations without invoking fault tolerance?
- Noise-tolerant algorithms

Quantum simulation as an algorithmic tool

- Linear algebra in Hilbert space: linear systems, differential equations, convex optimization, ...
- Find new applications of quantum simulation