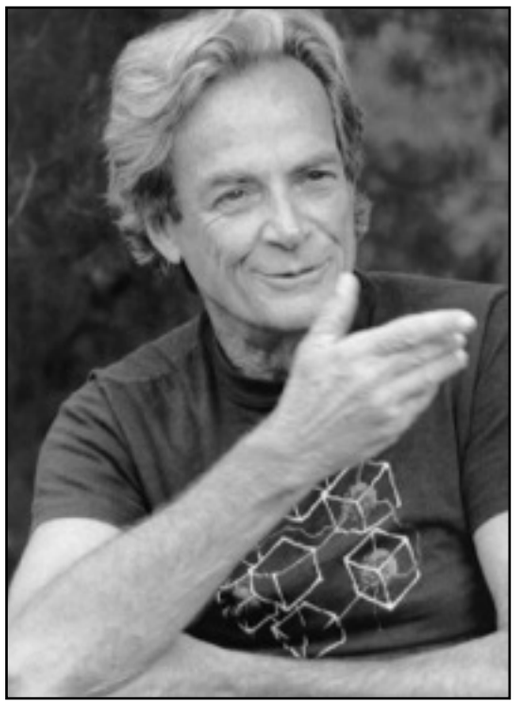
Simulating Hamiltonian dynamics on a small quantum computer

Andrew Childs

Department of Combinatorics & Optimization
and Institute for Quantum Computing
University of Waterloo

based in part on joint work with

Dominic Berry, Richard Cleve,
Robin Kothari, and Rolando Somma



"... 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)

Outline

- The Hamiltonian simulation problem
- Three approaches to simulation
  - Product formulas
  - Quantum walk
  - Fractional queries
- Comparison of methods for small-scale simulations

Hamiltonian simulation

Problem: Given* a Hamiltonian H, a time t, and an error tolerance ϵ (say, with respect to trace distance), find a quantum circuit that performs the unitary operation e^{-iHt} (on an unknown quantum state) with error at most ϵ.

* For an efficient simulation, H should be concisely specified.

More generally, H can be time-dependent.

Applications:

- Simulating physics
- Implementing continuous-time quantum algorithms (quantum walk, adiabatic optimization, linear equations, ...)

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)

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

What should we simulate?

Consider a two-dimensional grid of n qubits, with interactions only between nearest neighbors (at most $2n$ terms).

- Initialize system in a product state
- Evolve for time t
- Measure a local observable for each spin

This is probably hard to do with a classical computer for more than about 30 qubits.

Related question: How can we apply quantum simulation to solve computational problems in quantum chemistry, materials science, etc.? (See, e.g., the talks by Alán and Matthias.)

Overview of simulation methods

Product formulas

- Decompose Hamiltonian into a sum of terms and recombine by alternating between them
- With high-order formulas, complexity can be close to linear in t, but with an exponential prefactor

Quantum walk

- Implement a unitary operation whose spectrum is related to the Hamiltonian; use phase estimation to adjust the eigenvalues
- Complexity is linear in t with no large prefactor; also improved dependence on sparsity

Fractional queries

- Perform a compressed simulation of a very accurate product formula
- Strictly improves over product formulas, with no large prefactors
- Dependence on error is $\mathrm{poly}(\log(1/\epsilon))$ instead of $\mathrm{poly}(1/\epsilon)$

Product formulas

Simulating a sum of terms

Suppose we want to simulate $H = \sum_{i=1}^{m} H_i$

Combine individual simulations with the Lie product formula:

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

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

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

[Lloyd 96]

High-order product formulas

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

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

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

$$\vdots$$

Systematic expansions to arbitrary order are known [Suzuki 92]

Using the kth order expansion, the number of exponentials required for an approximation with error at most ϵ is at most

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

With k large, this is only slightly superlinear in t. Sublinear simulation is impossible ("no-fast-forwarding theorem").

[Berry, Ahokas, Cleve, Sanders 07]

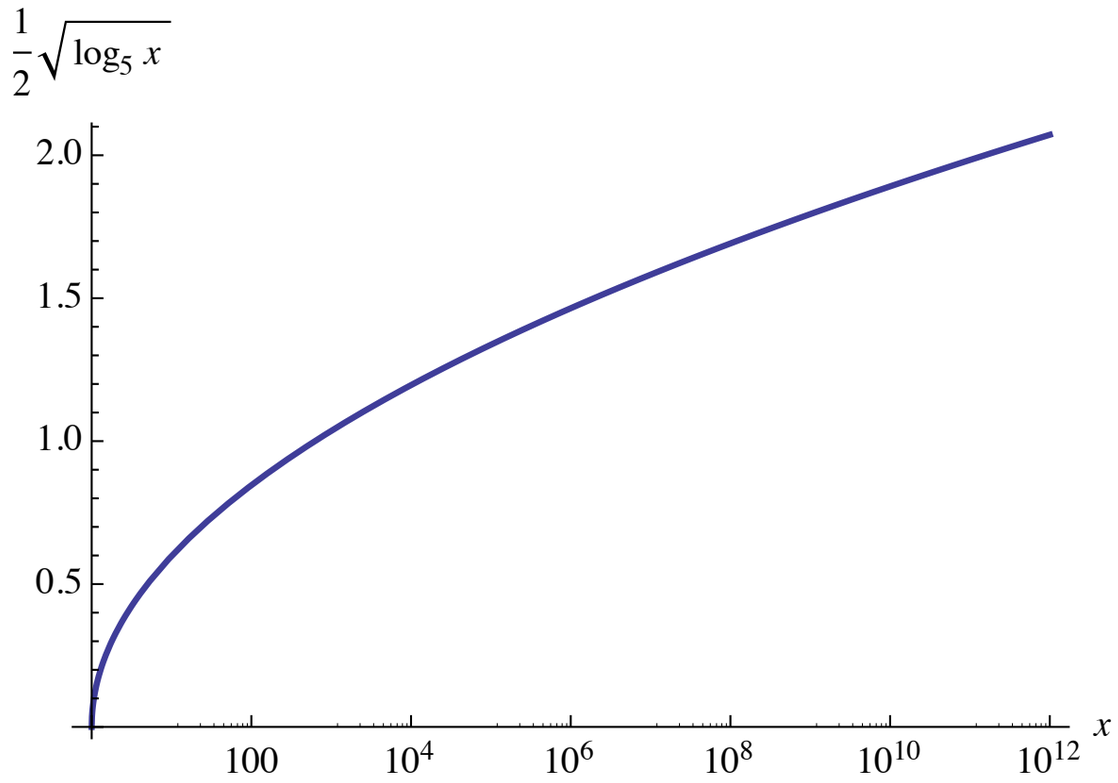
High-order product formulas

Choose k to minimize $5^{2k} m^2 \|Ht\| \left(\frac{m\|H\|t}{\epsilon} \right)^{1/2k}$

$$k \approx \frac{1}{2} \sqrt{\log_5 \left(\frac{m\|H\|t}{\epsilon} \right)} \quad \text{gives} \quad O\left(m^2 \|H\| t \, 5^{2\sqrt{\log_5 (m\|H\|t/\epsilon)}} \right)$$

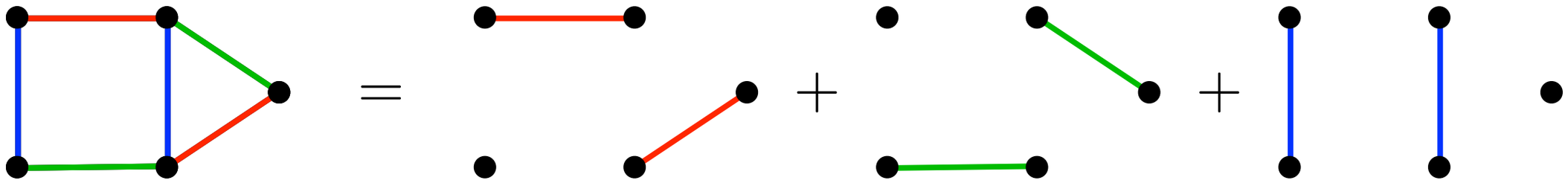
This is subpolynomial (but superlogarithmic) in $1/\epsilon$; closer to polynomial (cf. best known classical factoring algorithms).

When does it help to use higher-order formulas?



Sparse Hamiltonians and coloring

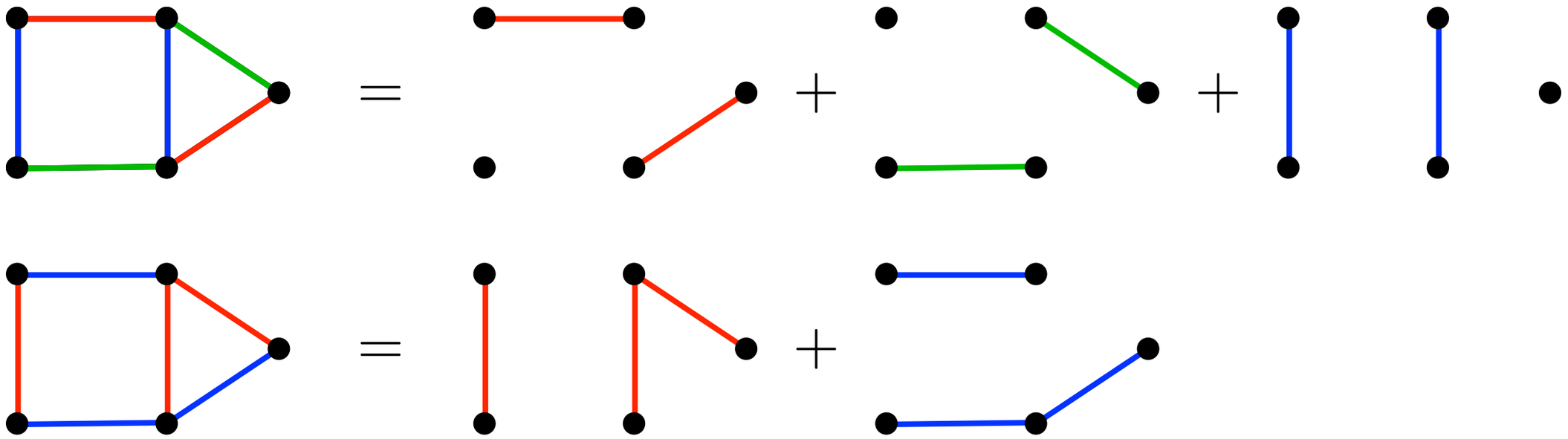
Strategy [Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 03; Aharonov, Ta-Shma 03]: Color the edges of the graph of H. Then the simulation breaks into small pieces that are easy to handle.



A sparse graph can be efficiently colored using only local information [Linial 87], so this gives efficient simulations.

Can produce a d^2-coloring locally $\Rightarrow$ overhead of d^4

Star decompositions



Strategy: Color the edges so that each color forms a "galaxy" (every component is a star graph). Simulate each galaxy by brute force and recombine.

Tradeoff vs. edge coloring:
- Decomposition has fewer terms
- Each term is harder to simulate (2nd neighbors)

Total overhead is d^3

[Childs, Kothari 10]

Quantum walk

[Childs, arXiv:0810.0312]

[Berry and Childs, arXiv:0910.4157]

Hamiltonian simulation by quantum walk

Another way to simulate an $N \times N$ Hamiltonian H is to implement a related discrete-time (Szegedy) quantum walk.

Expand space from $\mathbb{C}^N$ to $\mathbb{C}^{N+1} \otimes \mathbb{C}^{N+1}$.

Walk operator is the product of two reflections:

- Swap: $S|j,k\rangle = |k,j\rangle$
- Reflect about $\mathrm{span}\{|\psi_1\rangle, \ldots, |\psi_N\rangle\}$, where

$$|\psi_j\rangle := |j\rangle \otimes \left(\frac{1}{\sqrt{\|H\|_1}} \sum_{k=1}^{N} \sqrt{H_{jk}^*}\, |k\rangle + \nu_j |N+1\rangle \right)$$

$$\|H\|_1 := \max_j \textstyle\sum_{k=1}^{N} |H_{jk}|$$

i.e., $2TT^\dagger - \mathbb{1}$ where $T|j\rangle = |\psi_j\rangle$

[Childs 10]

Hamiltonian simulation by quantum walk

To simulate H for time t:

1. Apply T to the input state $|\psi\rangle$.
2. Perform phase estimation with $U = iS(2TT^\dagger - \mathbb{1})$, estimating a phase $\pm e^{\pm i \arcsin \lambda}$ for the component of $T|\psi\rangle$ corresponding to an eigenvector of H with eigenvalue λ.
3. Use the estimate of $\arcsin \lambda$ to estimate λ, and apply the phase $e^{-i\lambda t}$.
4. Uncompute the phase estimation and T, giving an approximation of $e^{-iHt}|\psi\rangle$.

Theorem: $O(\|Ht\|_1/\epsilon)$ steps of this walk suffice to simulate H for time t with error at most ϵ (in trace distance).

[Childs 10]

Faster simulation of sparse Hamiltonians

Perform quantum walk steps by brute force (query all d neighbors):

$$O(d\|H\|_1 t/\epsilon) \leq O(d^{3/2}\|H\|t/\epsilon) \leq O(d^2\|H\|_{\max} t/\epsilon)$$

This is exactly linear in t; also scales better in d, but worse in ϵ.

Improved alternative: using only two queries, prepare

$$|\psi'_j\rangle := |j\rangle \otimes \frac{1}{\sqrt{d}} \sum_{k=1}^{N} \left(\sqrt{\frac{H^*_{jk}}{\|H\|_{\max}}} |k,0\rangle + \sqrt{1 - \frac{|H_{jk}|}{\|H\|_{\max}}} |k,1\rangle \right)$$

$$\|H\|_{\max} := \max_{j,k} |H_{jk}|$$

Resulting complexity:

$$O\left(\frac{\|H\|t}{\sqrt{\epsilon}} + d\|H\|_{\max} t \right) \leq O\left(\frac{d\|H\|_{\max} t}{\sqrt{\epsilon}} \right)$$

[Berry, Childs 12]

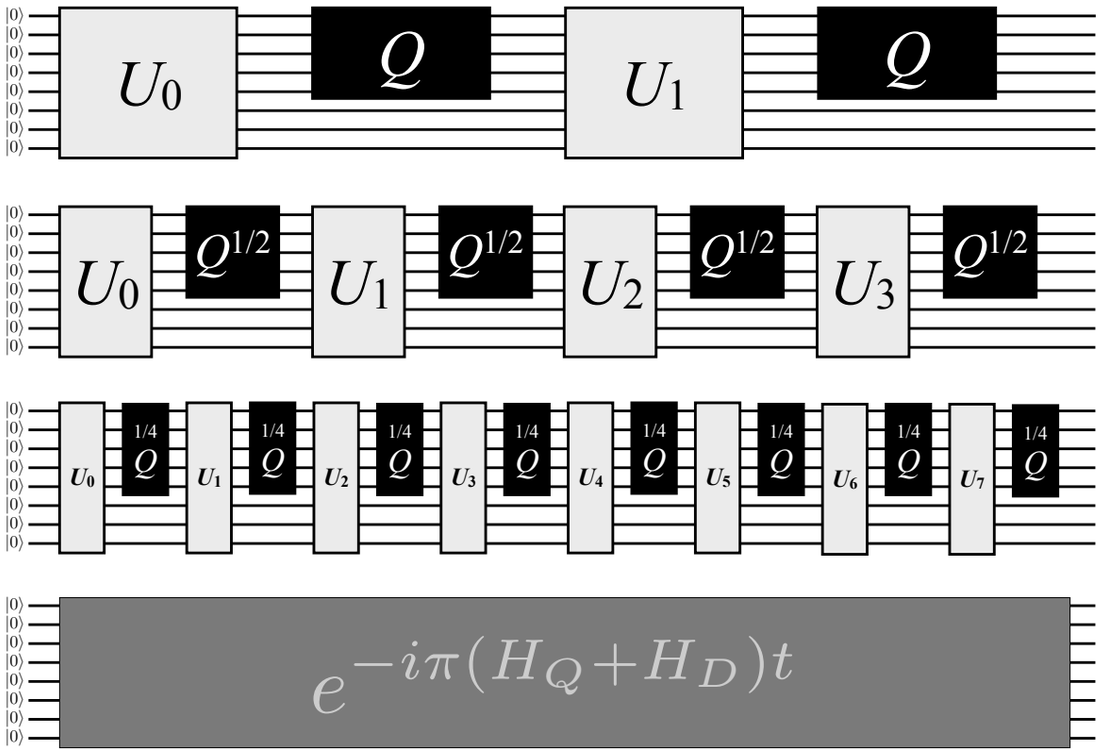
Fractional queries

[Berry, Childs, Cleve, Kothari, and Somma, arXiv:1312.1414]

Fractional- and continuous-query models

Black box hides a string $x \in \{0,1\}^n$

Quantum query: $Q|i,b\rangle = (-1)^{bx_i}|i,b\rangle$



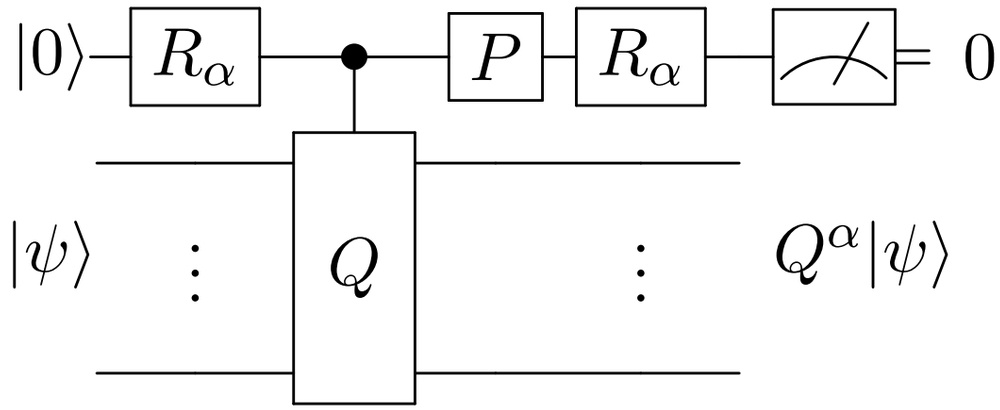
Useful for designing algorithms [Farhi, Goldstone, Gutmann 07]

More powerful than the discrete query model?

No: Can simulate a t-query fractional-query algorithm with $O(t\frac{\log t}{\log\log t})$ discrete queries [Cleve, Gottesman, Mosca, Somma, Yonge-Mallo 09]

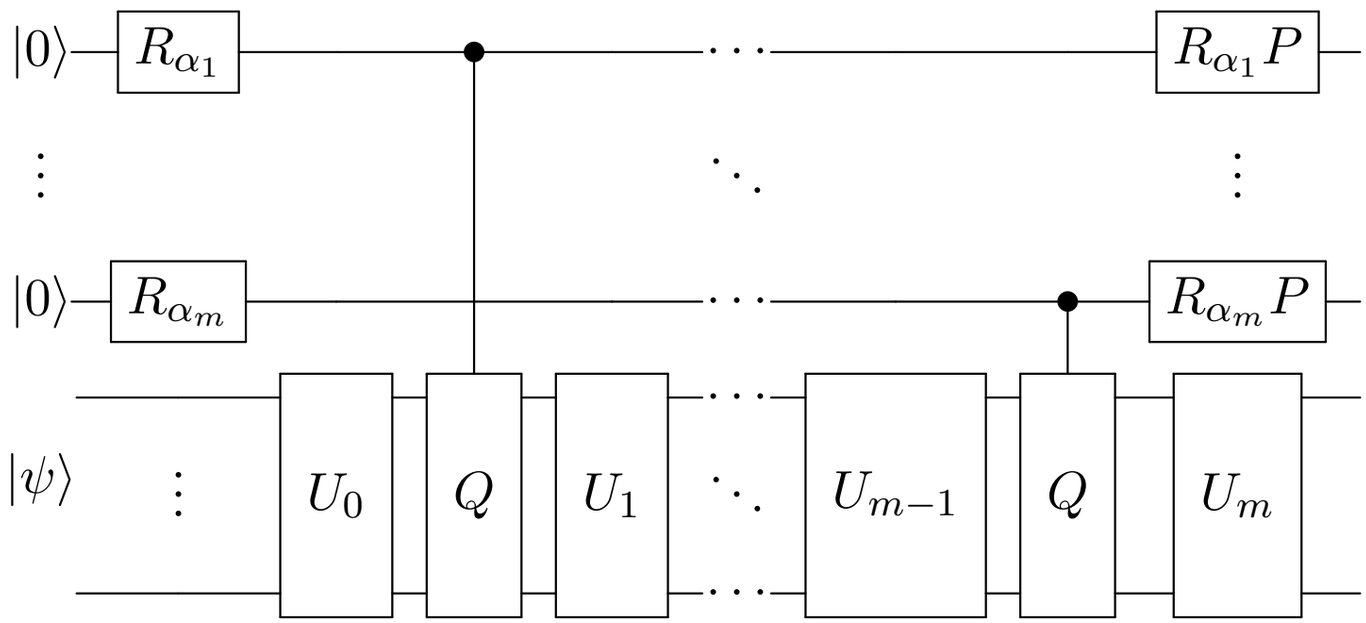
Simulating fractional queries

Fractional-query gadget:



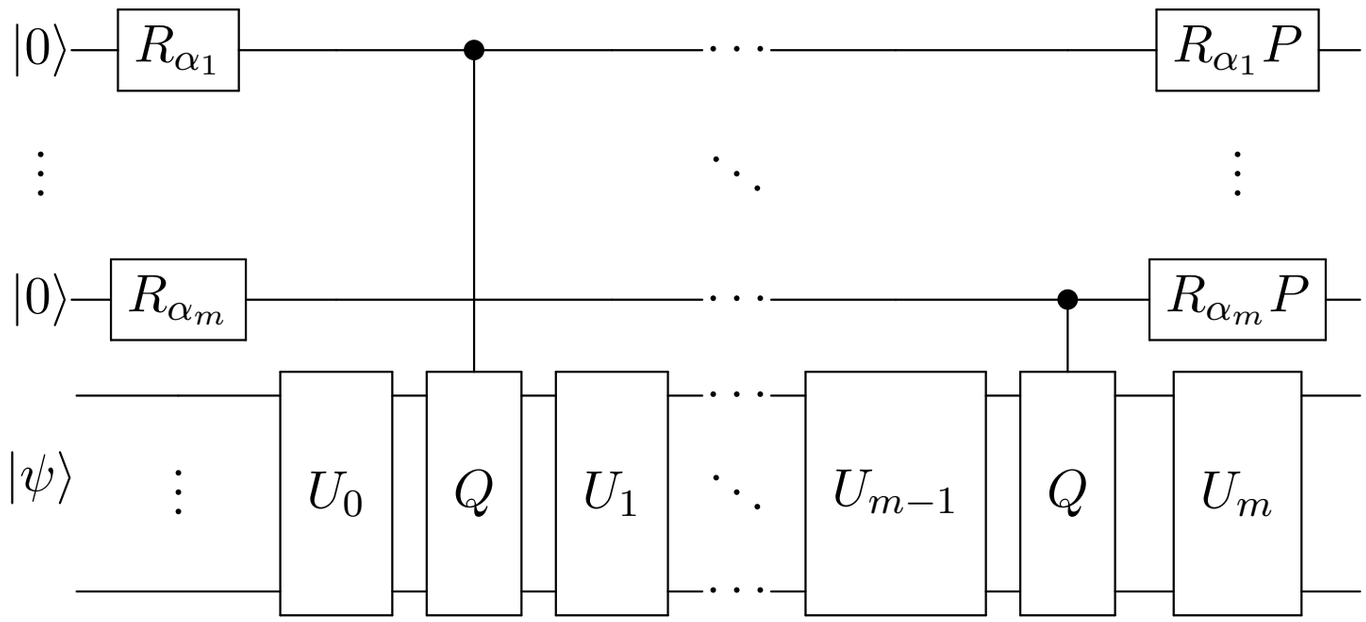
$$R_\alpha = \frac{1}{\sqrt{c+s}}\begin{pmatrix}\sqrt{c} & \sqrt{s}\\ \sqrt{s} & -\sqrt{c}\end{pmatrix} \qquad \begin{aligned} c &= \cos\tfrac{\pi\alpha}{2}\\ s &= \sin\tfrac{\pi\alpha}{2}\end{aligned} \qquad P = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}$$

"Segment" implementing $U_m Q^{\alpha_m} U_{m-1} \cdots U_1 Q^{\alpha_1} U_0$:



Behavior of a segment

"Segment" implementing $U_m Q^{\alpha_m} U_{m-1} \cdots U_1 Q^{\alpha_1} U_0$:

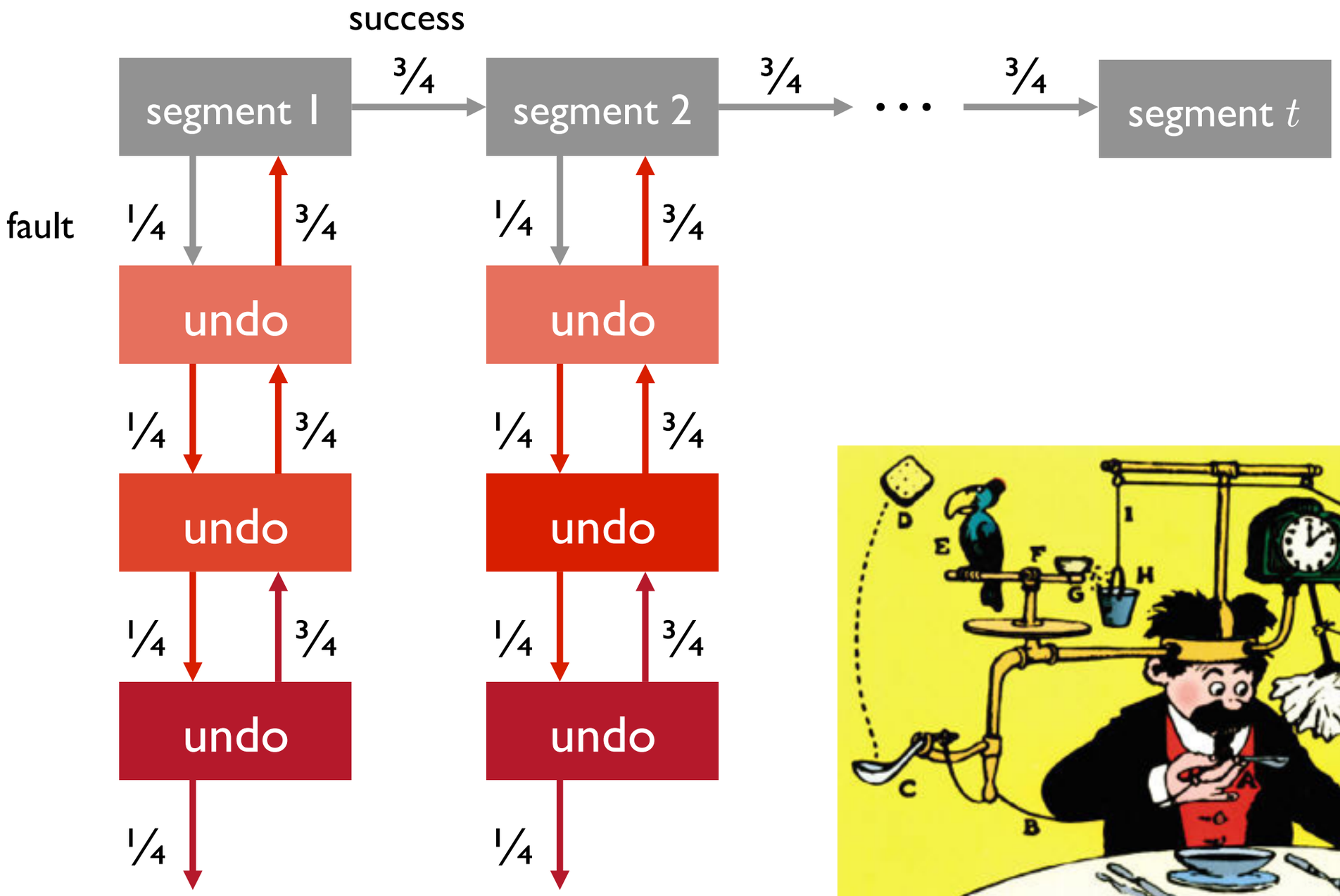


Truncating the ancillas to Hamming weight $k = O(\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)})$ introduces error at most ϵ

By rearranging the circuit, k queries suffice

But this still only succeeds with constant probability

Correcting faults



Rube Goldberg, *Professor Butts and the Self-Operating Napkin*

[Cleve, Gottesman, Mosca, Somma, Yonge-Mallo 09]

Oblivious amplitude amplification

Suppose U implements V with amplitude $\sin\theta$:

$$U|0\rangle|\psi\rangle = \sin\theta\,|0\rangle V|\psi\rangle + \cos\theta\,|1\rangle|\phi\rangle$$

segment (without final measurement)

$\frac{1}{2}$

ideal evolution

To perform V with amplitude close to 1: use amplitude amplification?

But the input state is unknown!

Using ideas from [Marriott, Watrous 05], we can show that a $|\psi\rangle$-independent reflection suffices to do effective amplitude amplification.

With this *oblivious amplitude amplification*, we can perform the ideal evolution *exactly* with only three segments (one backward).

Hamiltonian simulation using fractional queries

We reduce Hamiltonian simulation to fractional-query simulation.

Suppose $H = H_1 + H_2$ where H_1, H_2 have eigenvalues 0 and π. Write $e^{-i(H_1+H_2)t} \approx (e^{-iH_1 t/r} e^{-iH_2 t/r})^r$ for very large r (increasing r does not affect the query complexity, and only weakly affects the gate complexity).

This is a fractional-query algorithm with oracles e^{-iH_1} and e^{-iH_2}.

Package them as a single oracle $Q = |1\rangle\langle 1| \otimes e^{-iH_1} + |2\rangle\langle 2| \otimes e^{-iH_2}$.

This may not be diagonal in the standard basis, but the fractional-query simulation doesn't require that.

To give a complete simulation, we decompose the Hamiltonian into a sum of terms, each with eigenvalues 0 and π (up to an overall shift and rescaling). (Start by coloring the edges to make the Hamiltonian 1-sparse and then further refine the decomposition.)

Query and gate complexity

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

where $\tau := d^2 \|H\|_{\max} t$

Gate complexity is not much larger: $O\big(\tau \frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)} (\log(\tau/\epsilon) + n)\big)$

where H acts on n qubits

Local Hamiltonians

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

We can reduce the overhead below d^2 given a nice decomposition. Using the decomposition into m terms, each k-local (and hence 2^k-sparse), we can replace τ by $\tilde{\tau} := 2^k m \|H\|_{\max} t$.

Lower bounds

No-fast-forwarding theorem [BACS 07]: $\Omega(t)$

Main idea:

- Query complexity of computing the parity of n bits is $\Omega(n)$.
- There is a Hamiltonian that can compute parity by running for time $O(n)$.

New lower bound: $\Omega(\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)})$

Main idea:

- Query complexity of parity is $\Omega(n)$ even for *unbounded error*.
- The same Hamiltonian as above computes parity with unbounded error by running for any positive time. Running for constant time gives the parity with probability $\Theta(1/n!)$.

Comparison of sparse Hamiltonian simulations

	Product formulas	Quantum walk	Fractional queries
Query complexity	$d^3 \|\|H\|\| t \left(\frac{d\|\|H\|\| t}{\epsilon} \right)^{o(1)}$	$O\left(\frac{d\|\|H\|\|_{\max} t}{\sqrt{\epsilon}} \right)$	$O\left(\tau \frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\right)$ $\tau := d^2 \|\|H\|\|_{\max} t$
Best known scaling with evolution time t and sparsity d		✓	
Best known scaling with error ϵ			✓
Handles time-dependent Hamiltonians	✓		✓

How should we simulate small systems?

Consider a sum of m 2-local terms of constant norm acting on n qubits.

with $m = O(n)$

Product formulas

should use next-lowest order ($k = 1$)

gate complexity $O(m^{5/2} t^{3/2} / \sqrt{\epsilon})$ $O(n^{5/2} t^{3/2} / \sqrt{\epsilon})$

Quantum walk

gate complexity $O(mnt/\sqrt{\epsilon})$ $\tilde{O}(n^2 t/\sqrt{\epsilon})$

Fractional queries

gate complexity $O\left(mt \frac{\log(mt/\epsilon)}{\log\log(mt/\epsilon)} (\log(mt/\epsilon) + n)\right)$ $\tilde{O}(n^2 t)$

Open questions

- Improvements to methods; (optimal?) tradeoffs between evolution time, error, and locality/sparsity
- Careful estimates of gate complexity for small systems: what approach is best for particular simulations of interest?
- Investigate parallel algorithms
- Improved simulation of specific kinds of Hamiltonians
- Better understanding of applications to problems in quantum chemistry, etc.
- Real implementations!