Exponential improvement in precision for simulating sparse Hamiltonians

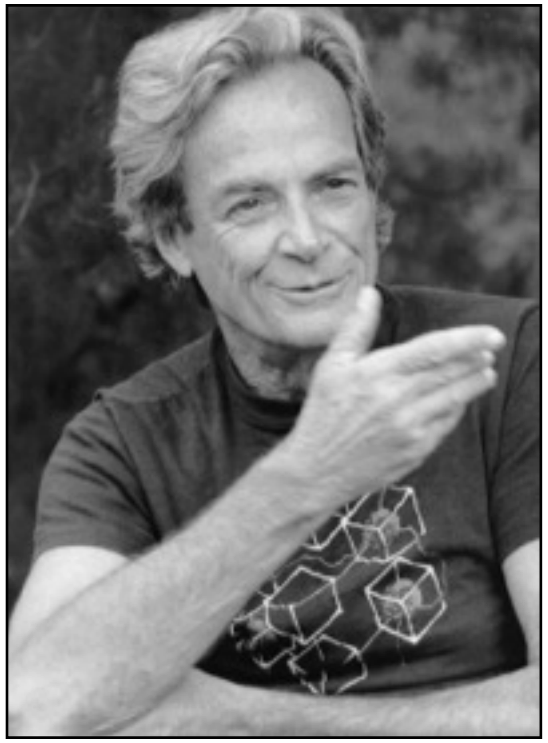
Andrew Childs

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

based on joint work with

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

arXiv:1312.1414
To appear in STOC 2014



"... 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
- adiabatic quantum computation
- linear equations

Quantum dynamics

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

$$i \frac{\mathrm{d}}{\mathrm{d}t} |\psi(t)\rangle = H |\psi(t)\rangle$$

Quantum simulation problem: Given a description of the Hamiltonian H, an evolution time t, and an unknown initial state $|\psi(0)\rangle$, produce the final state $|\psi(t)\rangle$ (approximately)

A classical computer cannot even represent the state efficiently

By performing measurements on the final state, a quantum computer can efficiently answer questions that (apparently) a classical computer cannot

Quantum simulation

Quantum simulation problem: Given a description of the Hamiltonian H, an evolution time t, and an unknown initial state $|\psi(0)\rangle$, produce the final state $|\psi(t)\rangle$ (approximately)

Equivalently, apply the unitary operator $U(t)$ satisfying

$$i\frac{\mathrm{d}}{\mathrm{d}t}U(t) = H(t)\,U(t)$$

If H is independent of t, $U(t) = e^{-iHt}$

More generally, H can be time-dependent

Simulation should approximate $U(t)$ to within error ϵ (say, with respect to the diamond norm)

Local and sparse Hamiltonians

Local Hamiltonians [Lloyd 96]

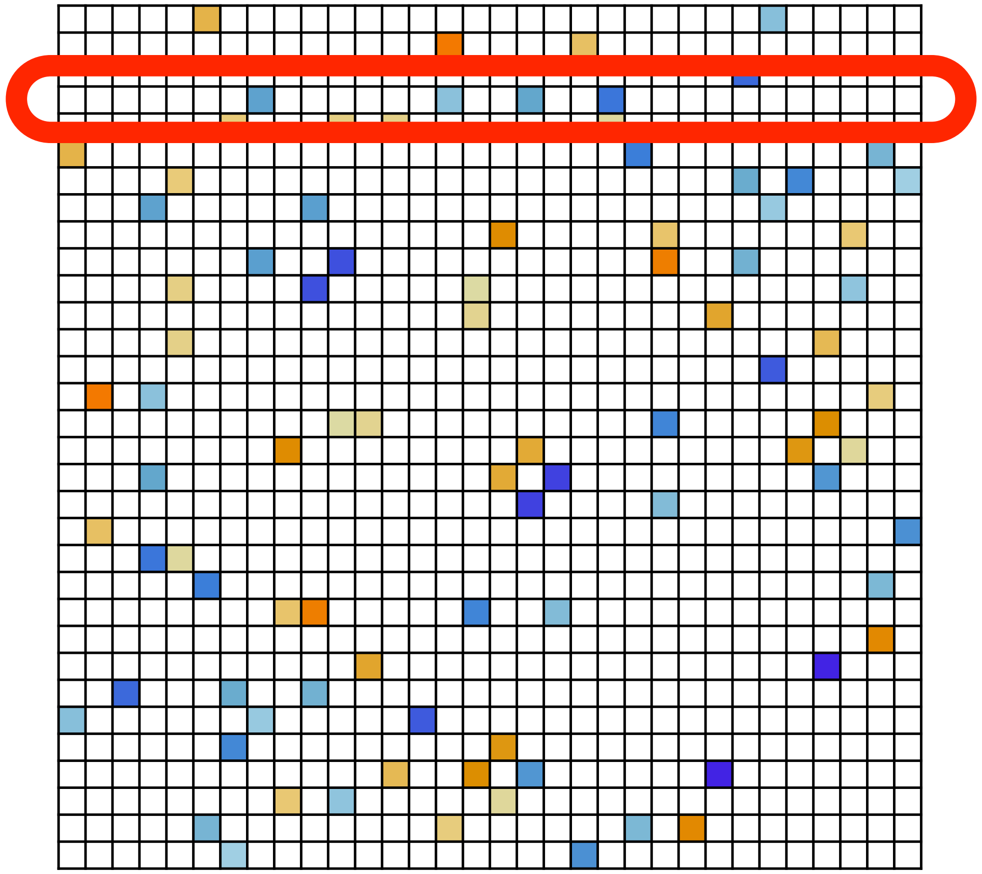
$H = \sum_{j=1}^{m} H_j$ where each H_j acts on $k = O(1)$ qubits

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 previous simulation methods

Parameters: dimension N sparsity d
evolution time t allowed error ϵ

[Lloyd 96]: $\mathrm{poly}(\log N)\,(\|H\|t)^2/\epsilon$ (for local Hamiltonians only)

[Aharonov, Ta-Shma 02]: $\mathrm{poly}(d, \log N)\,(\|H\|t)^{3/2}/\sqrt{\epsilon}$

[Childs 04]: $O\big((d^4 \log^4 N \|H\|t)^{1+\delta}/\epsilon^\delta\big)$ (for any $\delta > 0$)

[Berry, Ahokas, Cleve, Sanders 07]: $O\big((d^4 \log^* N \|H\|t)^{1+\delta}/\epsilon^\delta\big)$

[Childs, Kothari 11]: $O\big((d^3 \log^* N \|H\|t)^{1+\delta}/\epsilon^\delta\big)$

[Childs10; Berry, Childs 12]: $O(d\|H\|_{\max} t/\sqrt{\epsilon})$

New result: $O\big(\tau \frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\big)$ $\tau := d^2 \|H\|_{\max} t$

Fractional-query simulation

New approach: use tools for simulating the *fractional-query model*

Two steps:

- Reduce Hamiltonian simulation to fractional-query simulation
- Improved algorithm for fractional-query simulation

High-level idea of fractional-query simulation:

- Decompose the evolution into terms that can be implemented in superposition
- "Compress" the implementation
- Unit-time evolution only succeeds with constant probability; boost this using *oblivious amplitude amplification*

Strictly improves all methods based on product formulas

Dependence on ϵ is exponentially improved!
In fact, the improved dependence is optimal.

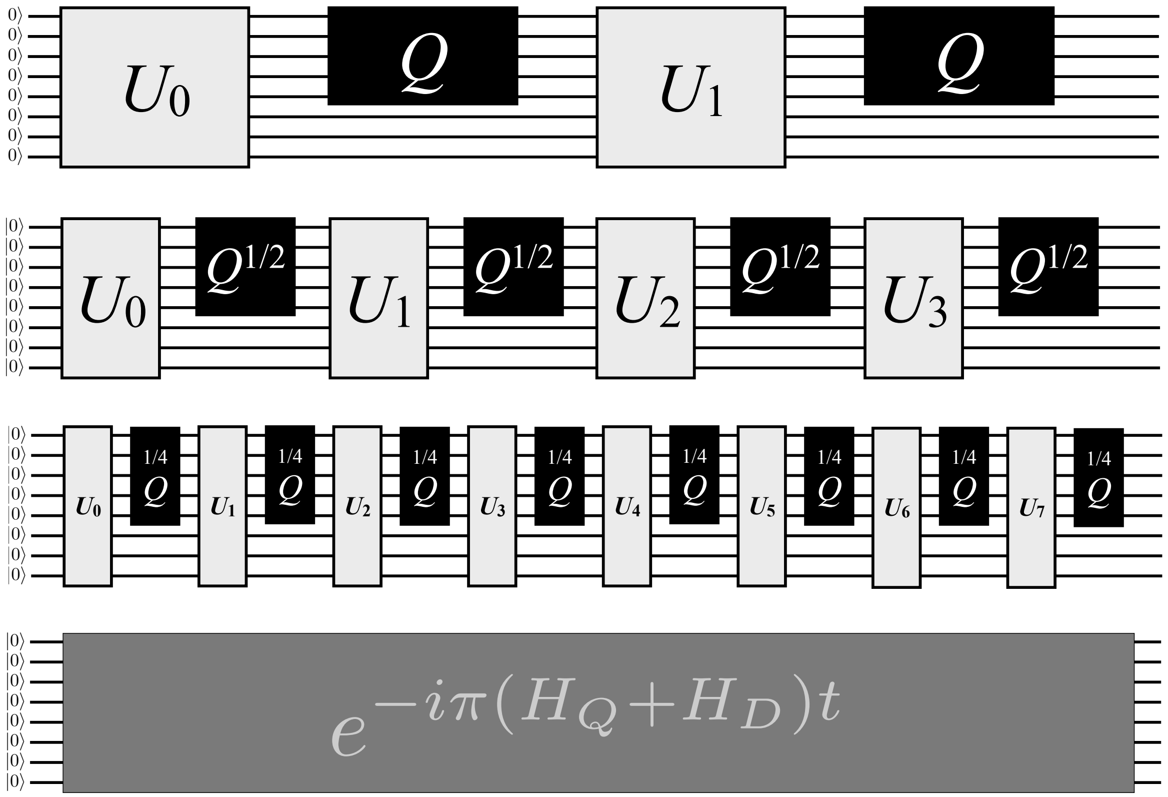
Outline

- Fractional-query model
  - Simulating fractional queries
  - Oblivious amplitude amplification
- Reducing Hamiltonian simulation to fractional-query simulation
- Features of the algorithm
  - Gate complexity
  - Local Hamiltonians
  - Time-dependent Hamiltonians
- Optimality with respect to error
- Comparison of simulation methods
- Open questions

Fractional- and continuous-query models

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

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



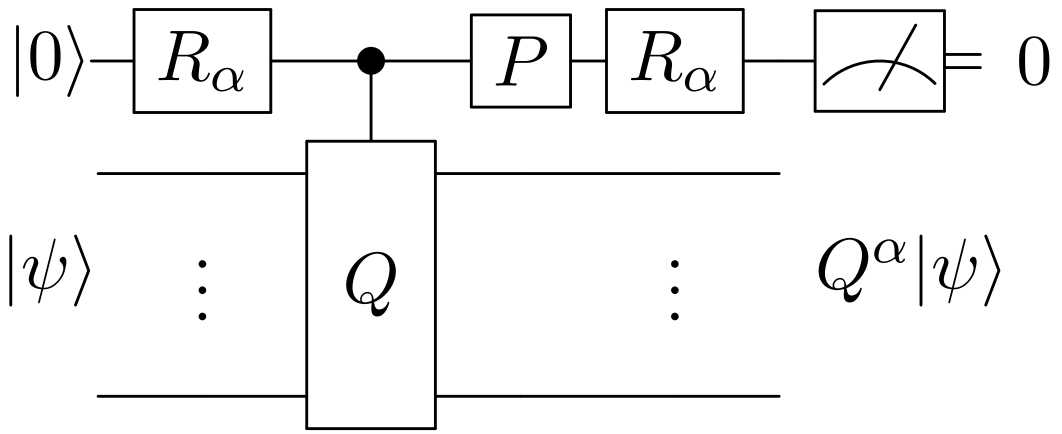
Useful for designing algorithms [Farhi, Goldstone, Gutmann 07]

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]

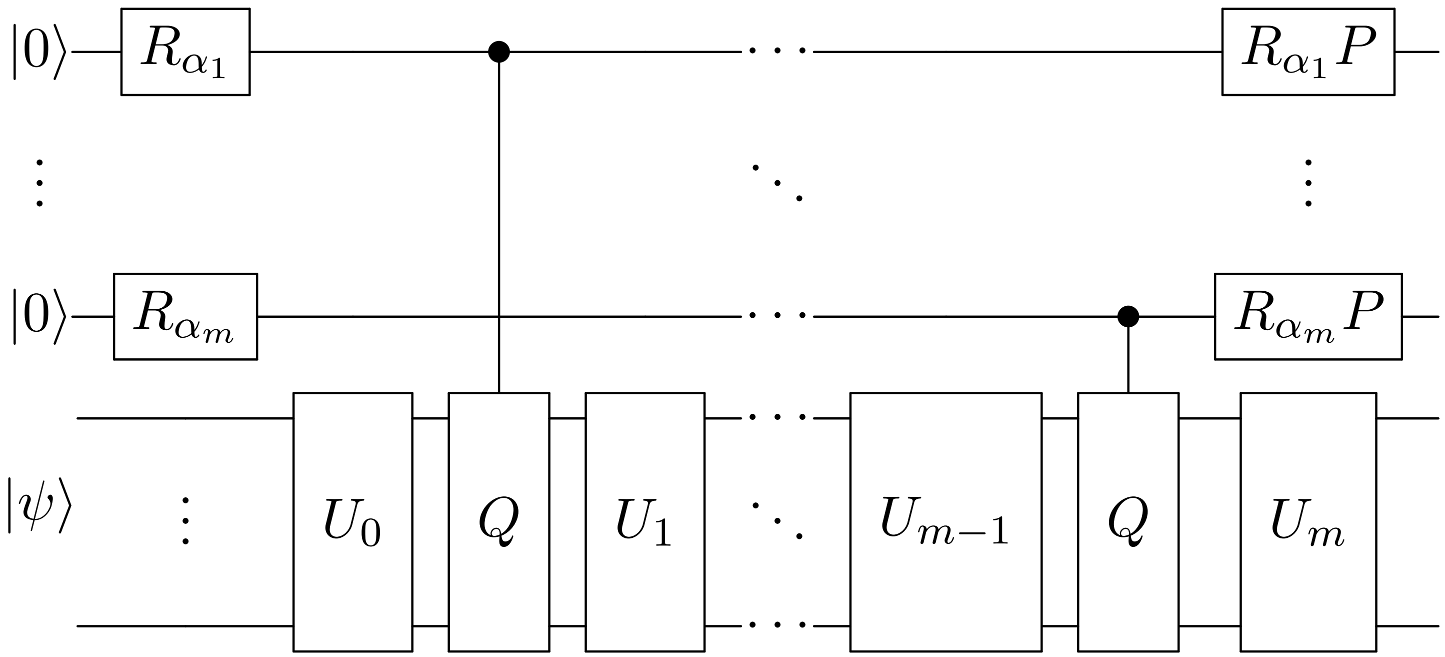
Simulating fractional queries

Fractional-query gadget:



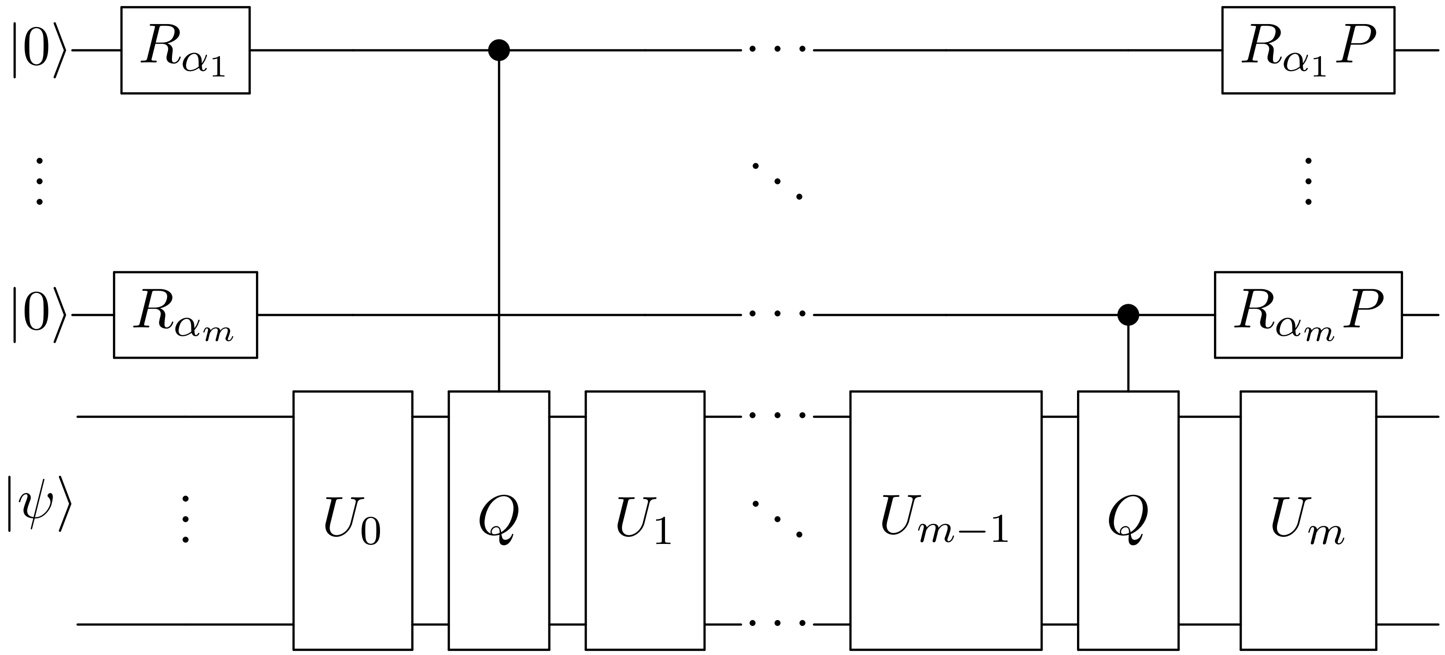
$$R_\alpha = \frac{1}{\sqrt{c+s}}\begin{pmatrix}\sqrt{c} & \sqrt{s} \\ \sqrt{s} & -\sqrt{c}\end{pmatrix} \qquad \begin{aligned} c &= \cos\tfrac{\pi\alpha}{2} \\ s &= \sin\tfrac{\pi\alpha}{2}\end{aligned} \qquad P = \begin{pmatrix}1 & 0 \\ 0 & i\end{pmatrix}$$

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



Behavior of a segment

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

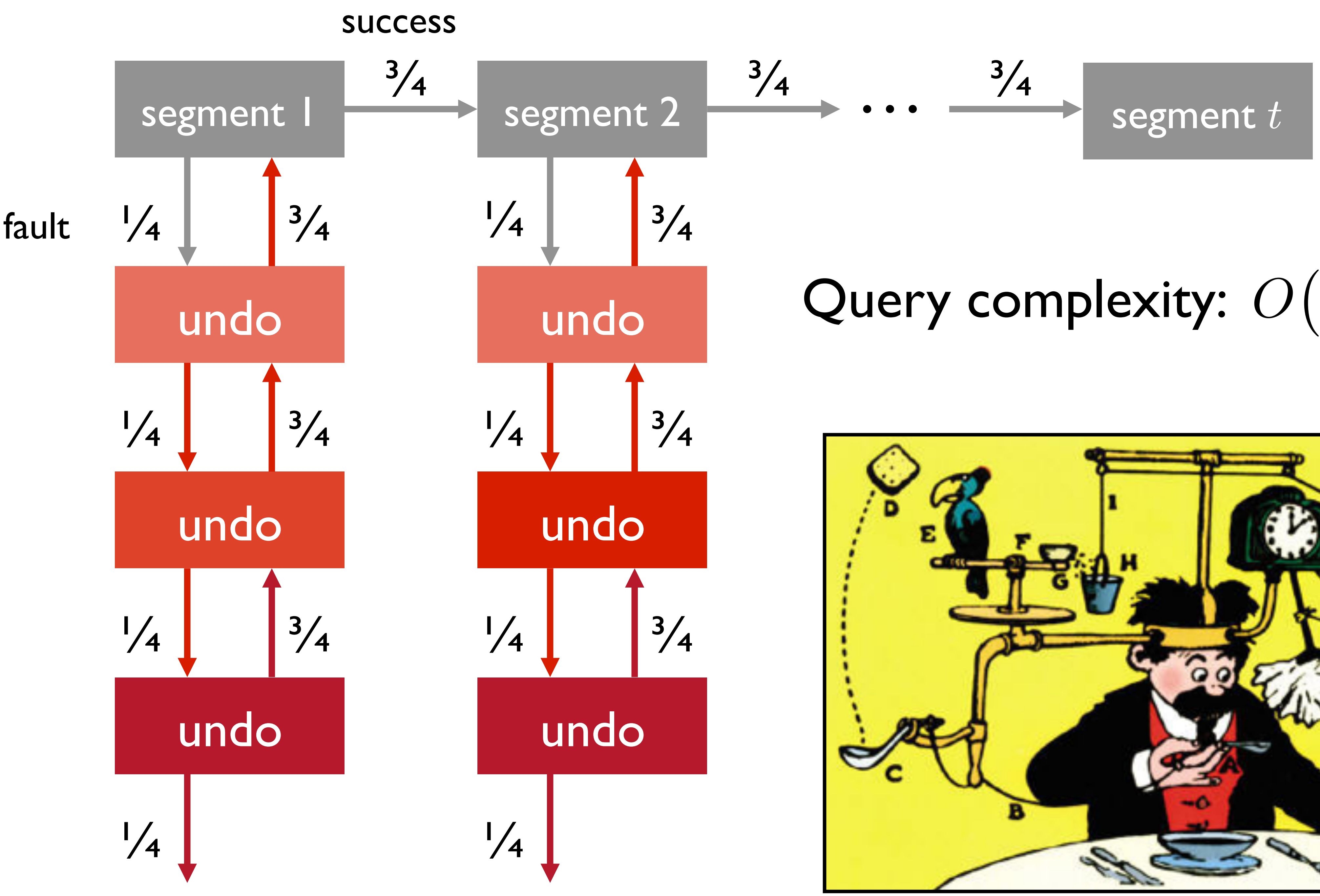


Truncating the ancillas to Hamming weight $k = O(\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)})$ introduces error at most ϵ

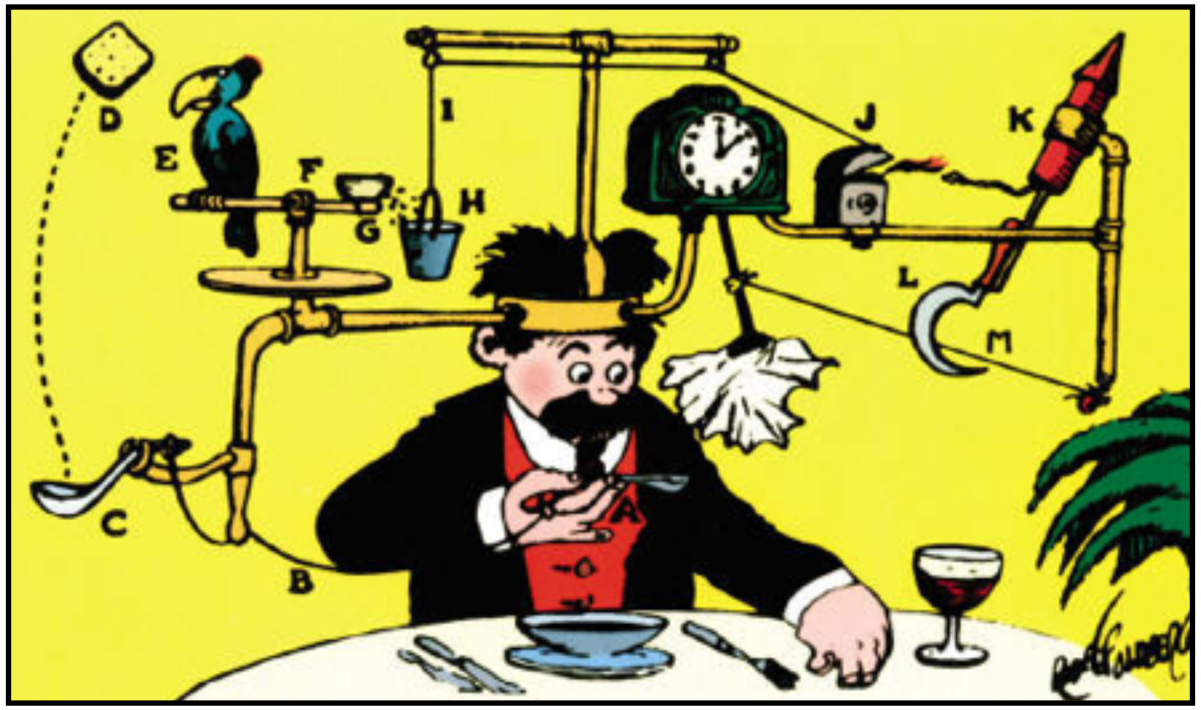
By rearranging the circuit, k queries suffice

But this still only succeeds with constant probability

Correcting faults



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



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 $Q = |1\rangle\langle 1| \otimes e^{-iH_1} + |2\rangle\langle 2| \otimes e^{-iH_2}$.

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

Decomposing sparse Hamiltonians

To give a complete simulation, decompose the d-sparse Hamiltonian into a sum of terms, each with eigenvalues 0 and π (up to an overall shift and rescaling).

- Edge coloring: $H = \sum_{j=1}^{d^2} H_j$ where each H_j is 1-sparse

  new trick: H is bipartite wlog since it suffices to simulate $H \otimes \sigma_x$

  d^2-coloring: $\mathrm{color}(\ell, r) = (\mathrm{idx}(\ell, r), \mathrm{idx}(r, \ell))$

- Approximately decompose into terms with all nonzero entries equal

Ex: $$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 3 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

- Remove zero blocks so that all terms have two fixed eigenvalues

Ex: $$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

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

Contributions to this gate complexity come from

- Preparing the ancilla state
- Performing simple operations between fractional queries
- Implementing the fractional-query oracle using the sparse Hamiltonian oracle

Using oblivious amplitude amplification instead of recursive fault correction considerably simplifies the analysis.

Local Hamiltonians

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

Directly applying our main result gives gate complexity

$$O\big(\tau \frac{\log^2(\tau/\epsilon)}{\log\log(\tau/\epsilon)} n\big) \qquad \tau := d^2 \|H\|_{\max} t = 4^k m^2 \|H\|_{\max} t$$

Ex: Generic 2-local Hamiltonian acting for constant time
$k = 2,\ m = \binom{n}{2},\ t, \|H\|_{\max} = O(1)$
gives complexity $\tilde{O}(n^5)$ (cf. high-order product formulas: $\tilde{O}(n^4)$)

But we can do better since we have an explicit decomposition into m k-local (and hence 2^k-sparse) terms. Resulting gate complexity:

$$O\big(\tilde{\tau} \frac{\log^2(\tilde{\tau}/\epsilon)}{\log\log(\tilde{\tau}/\epsilon)} n\big) \qquad \tilde{\tau} := 2^k m \|H\|_{\max} t$$

Ex: Generic 2-local Hamiltonian acting for constant time: $\tilde{O}(n^3)$

Time-dependent Hamiltonians

The query complexity of this approach depends only on the evolution time, not on the number of fractional-query steps

⇒ Query complexity of simulating sparse $H(t)$ is independent of $\|\frac{\mathrm{d}}{\mathrm{d}t}H(t)\|$ (provided this is bounded)

(cf. [Poulin, Quarry, Somma, Verstraete 11])

Gate complexity depends only weakly on $\|\frac{\mathrm{d}}{\mathrm{d}t}H(t)\|$:

$$O\Big(\tau\frac{\log(\tau/\epsilon)\log((\tau+\tau')/\epsilon)}{\log\log(\tau/\epsilon)}n\Big)$$

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

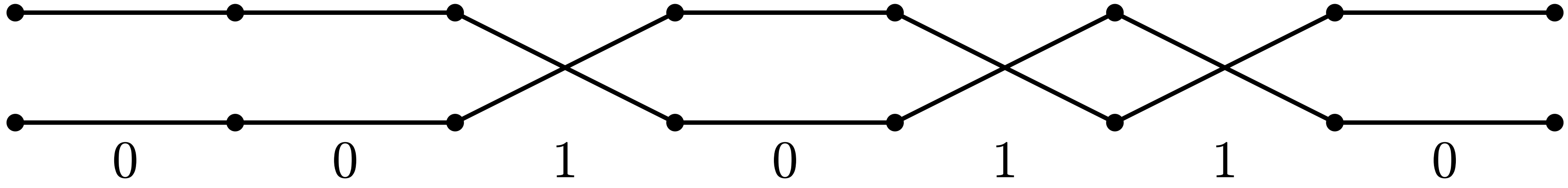
$$\tau' := d^2\max_{s\in[0,t]}\|\tfrac{\mathrm{d}}{\mathrm{d}s}H(s)\|\,t$$

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	✓		✓

Open questions

- Improvements to methods; (optimal?) tradeoffs between evolution time, error, and locality/sparsity
- Improved simulation of specific kinds of Hamiltonians
- Better understanding of applications to problems in quantum chemistry, etc.
- Performance for small systems; implementations