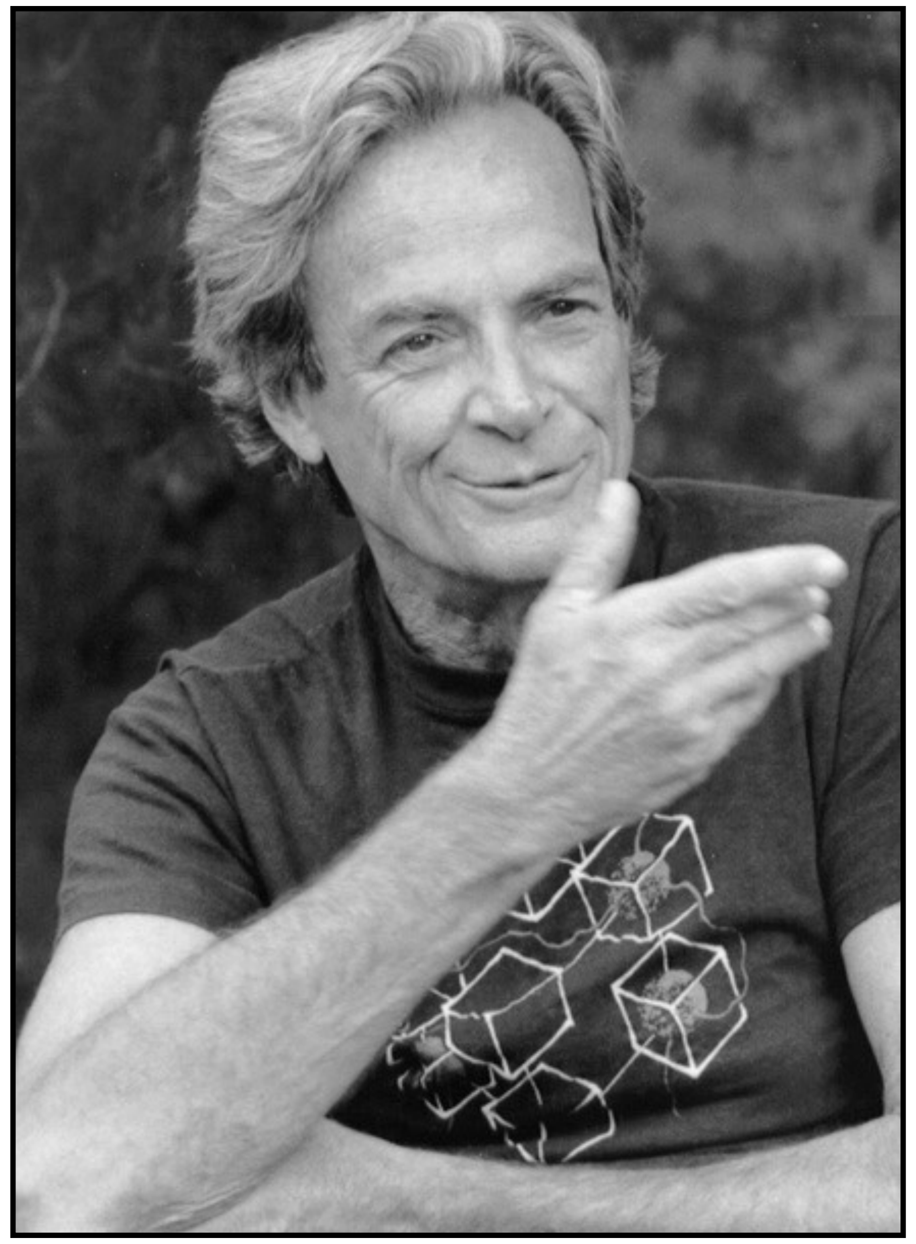
Algorithmic challenges in quantum simulation

Andrew Childs
University of Maryland

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

UMIACS
University of Maryland
Institute for Advanced
Computer Studies

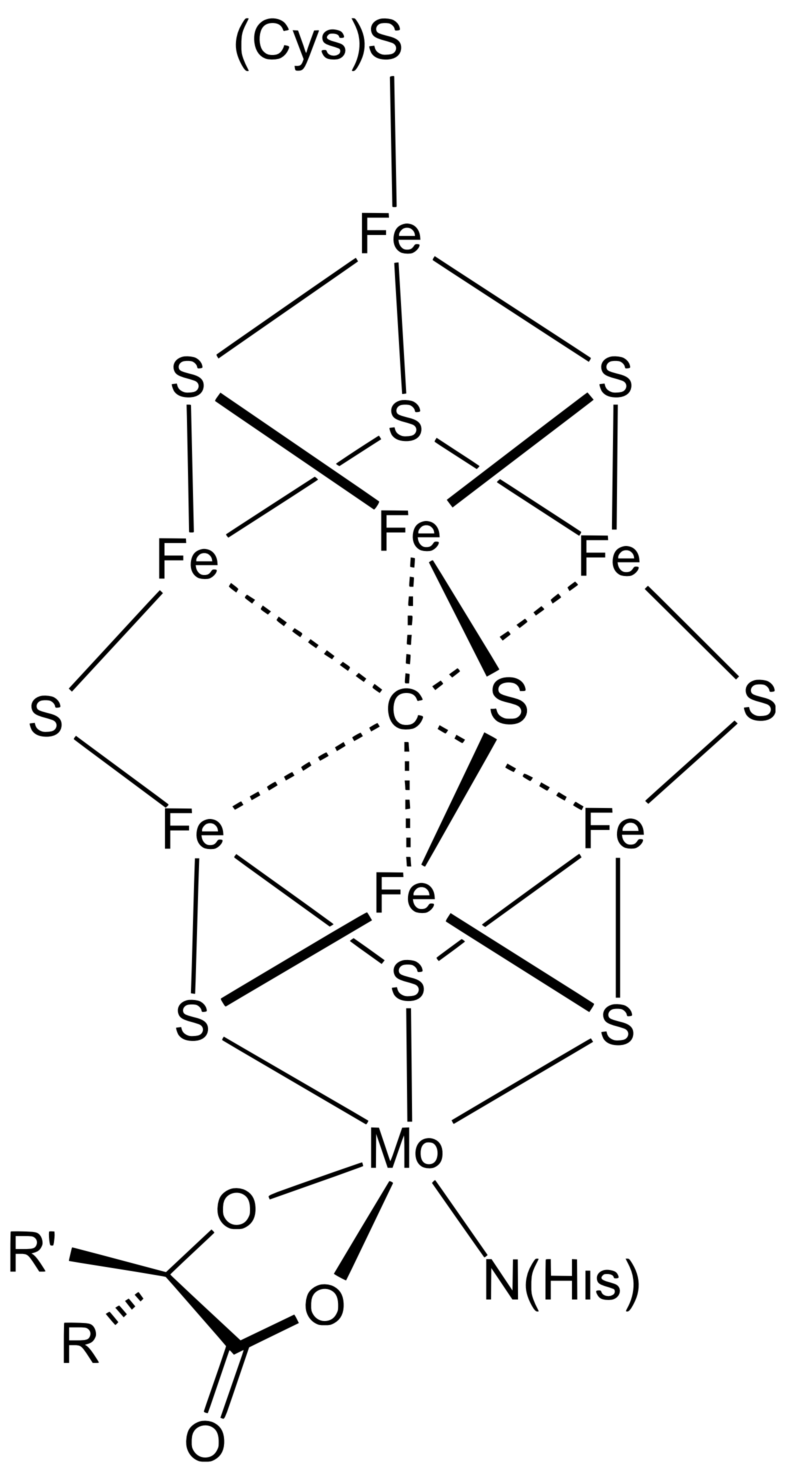
JOINT CENTER FOR
QUANTUM INFORMATION
AND COMPUTER SCIENCE



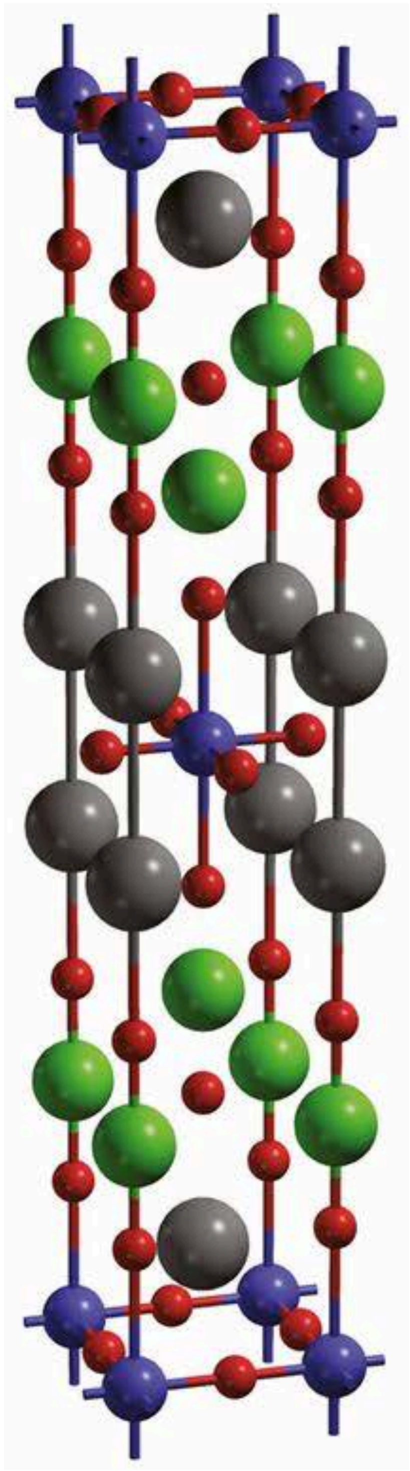
"… nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

Richard Feynman
Simulating physics with computers
MIT Physics of Computation Conference, 1981

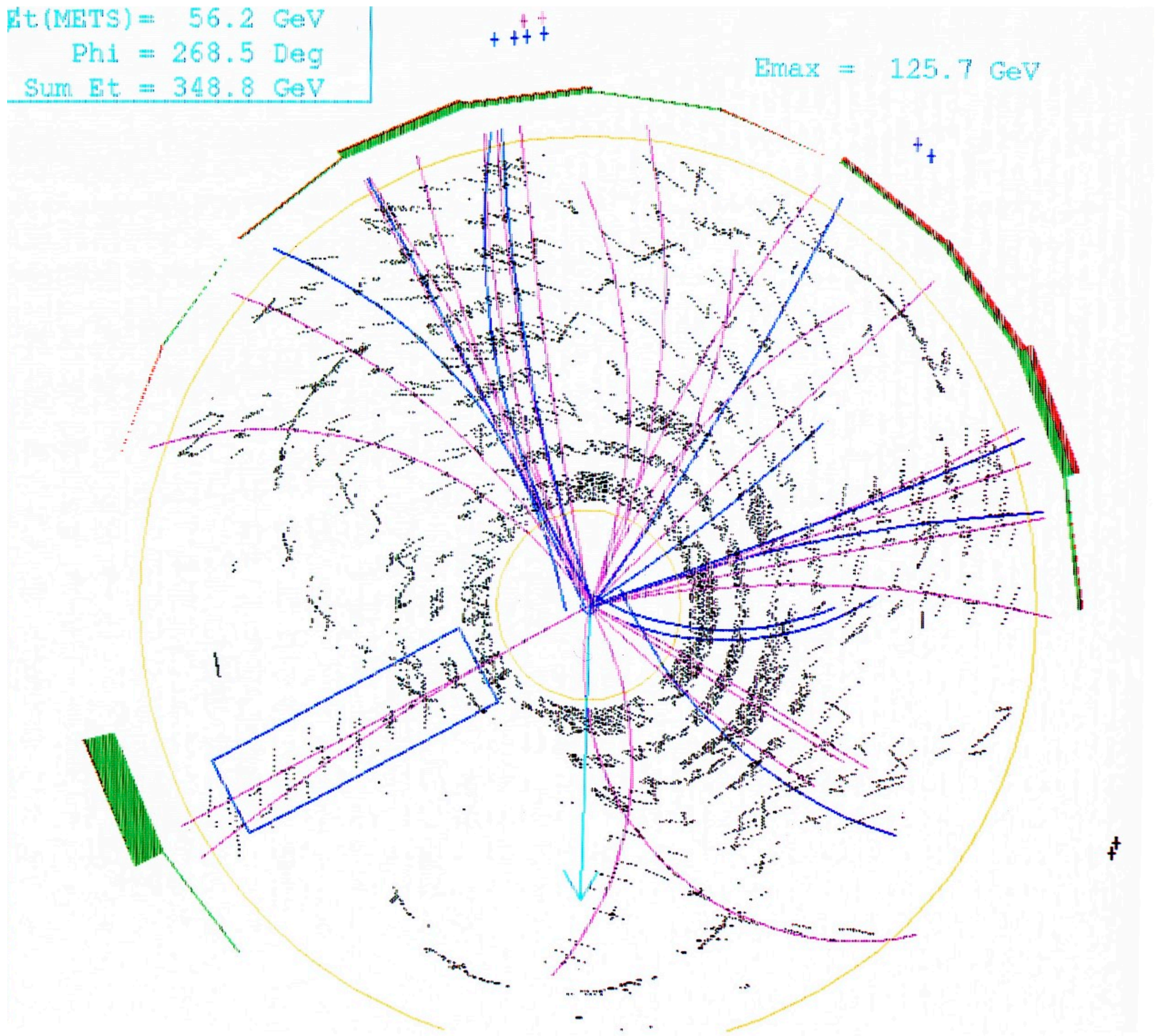
Computational quantum physics



chemical reactions
(e.g., nitrogen fixation)

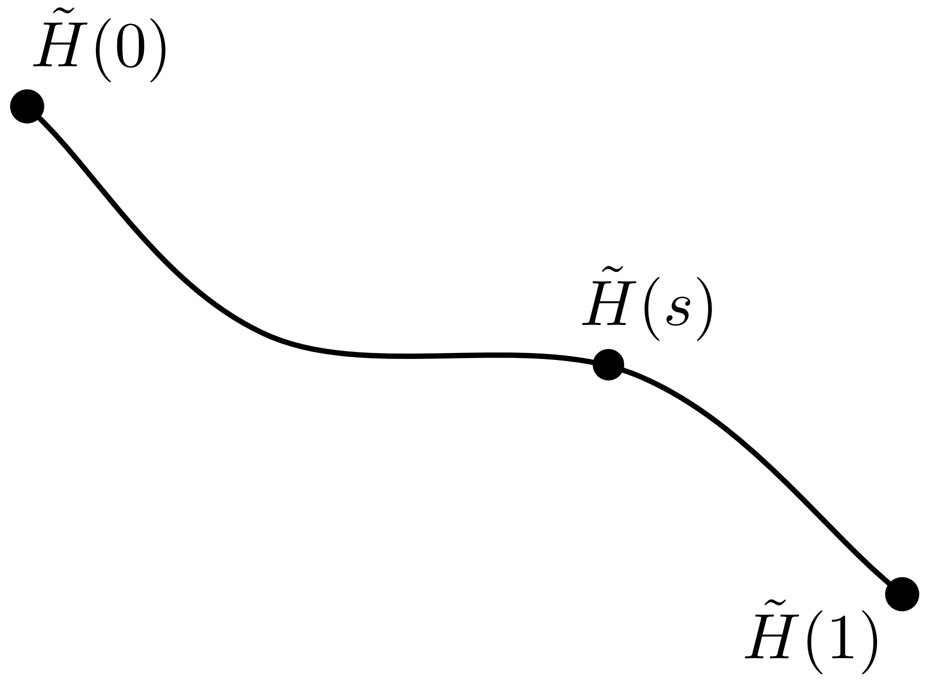


condensed matter physics/
properties of materials

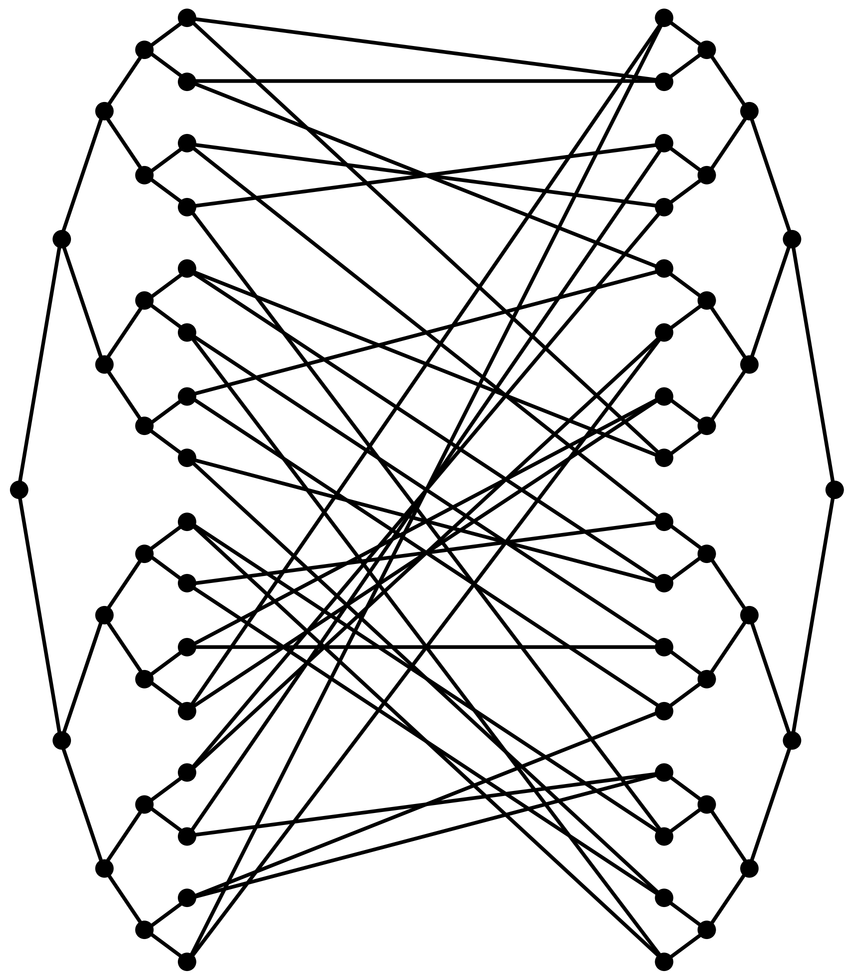


nuclear/particle
physics

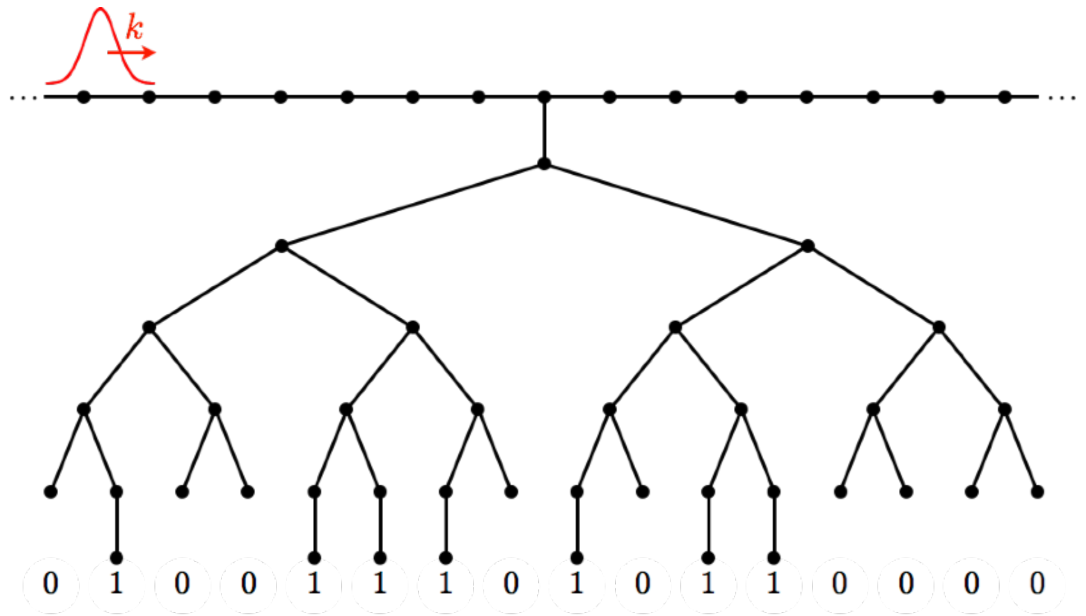
Implementing quantum algorithms



adiabatic optimization



exponential speedup by quantum walk



evaluating Boolean formulas

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

linear/ differential equations, convex optimization

Algorithmic challenges in quantum simulation

1. Efficient simulation with a universal quantum computer
2. Simulating quantum mechanics in real time
3. High-precision simulation
4. An optimal tradeoff
5. Real-time simulation revisited
6. Making quantum simulation practical

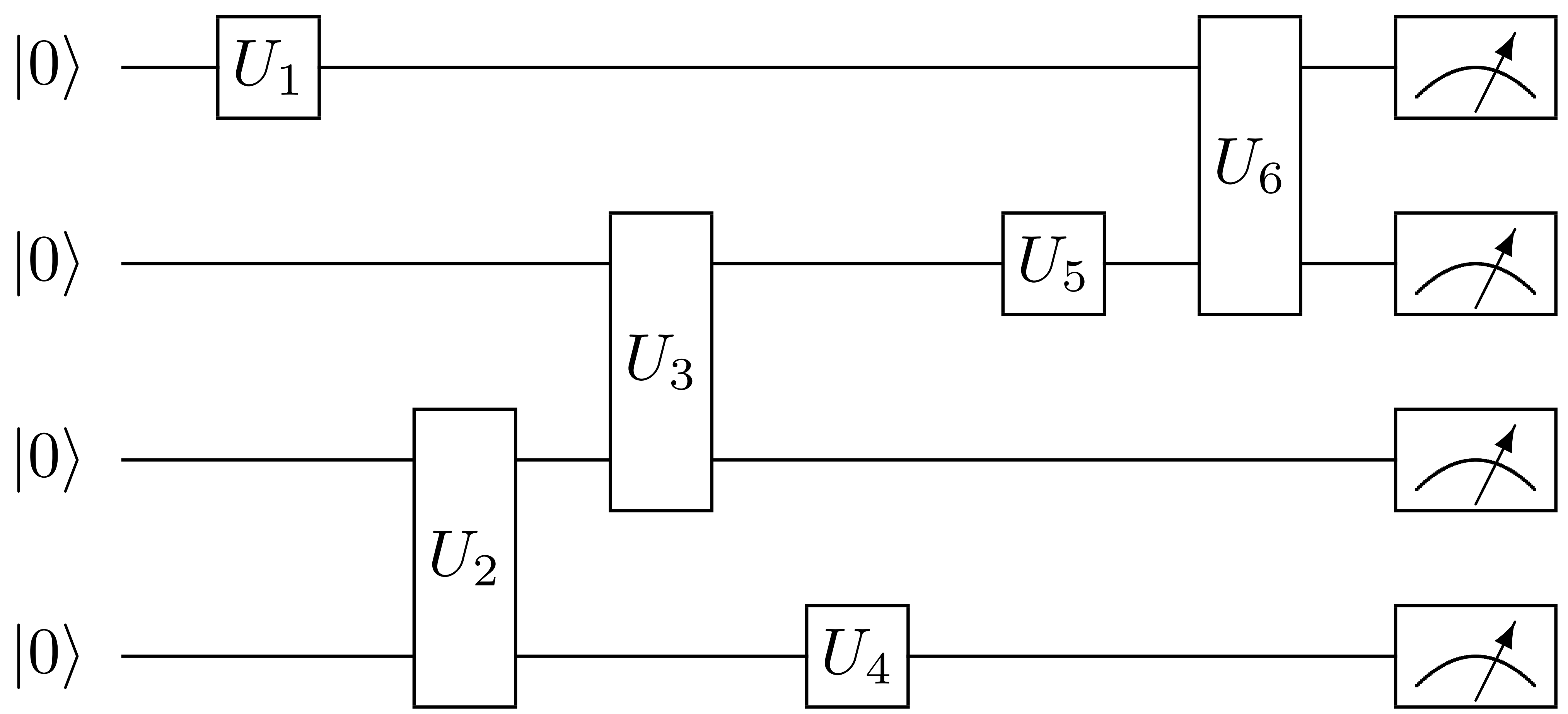
1. Efficient simulation with a universal quantum computer

1. Efficient simulation with a universal quantum computer

Universal quantum computation

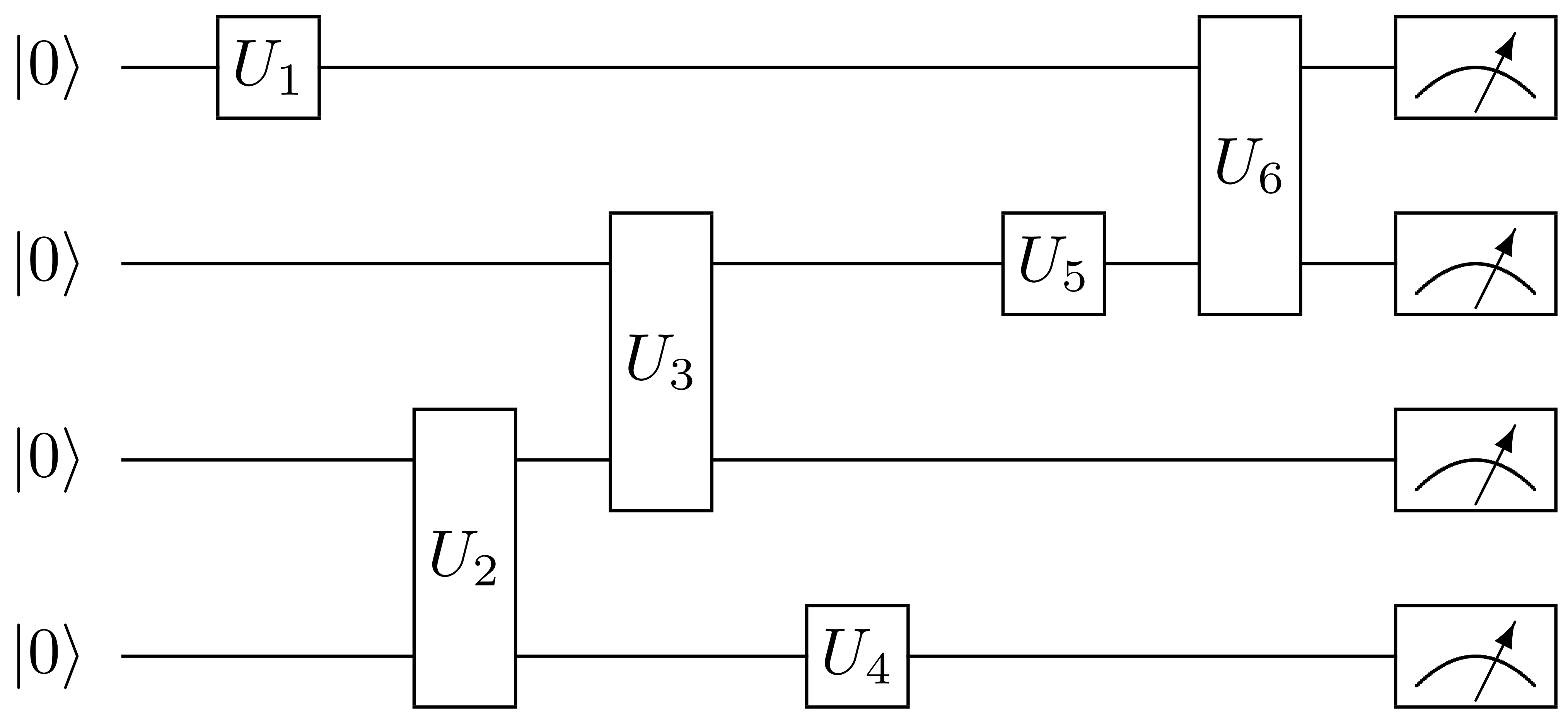
1. Efficient simulation with a universal quantum computer

Universal quantum computation



1. Efficient simulation with a universal quantum computer

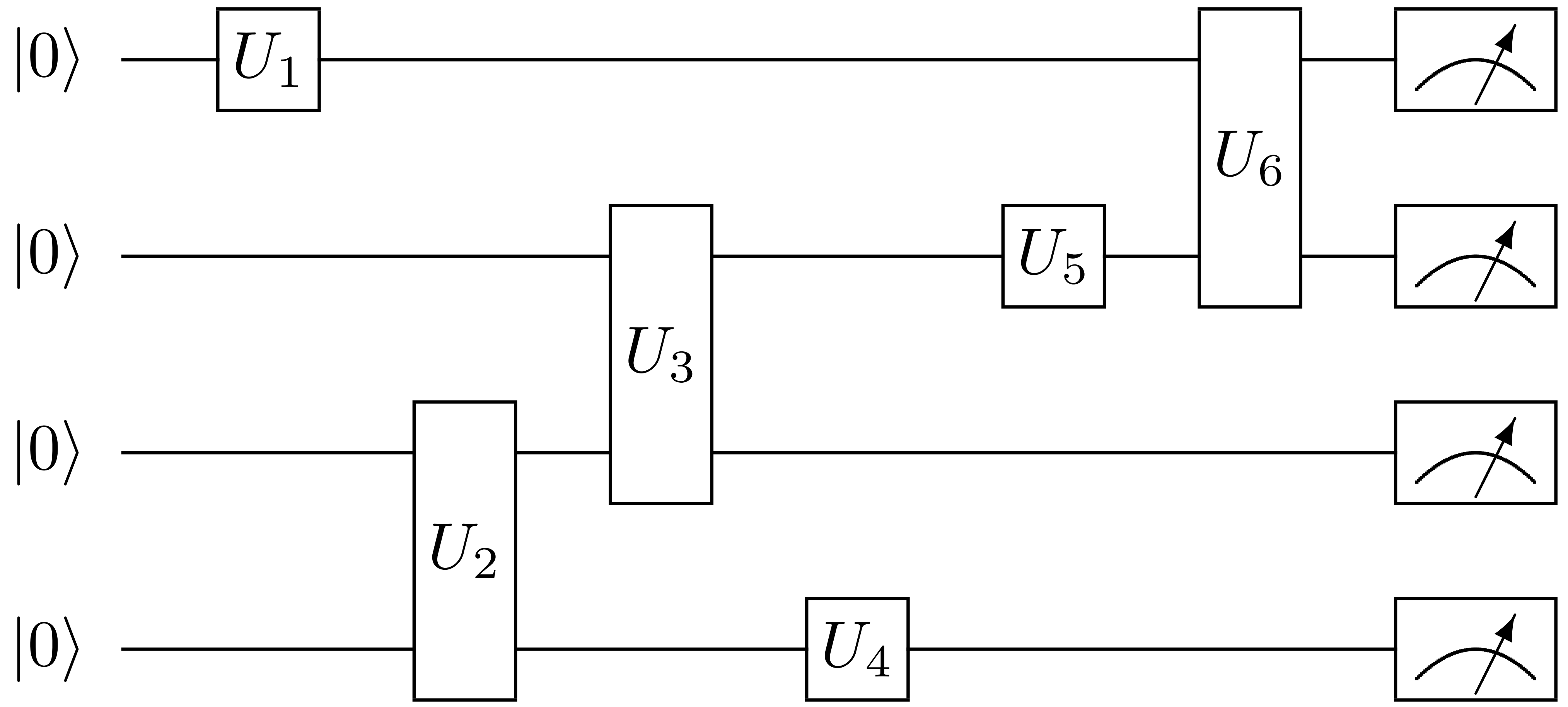
Universal quantum computation



The simulation problem

1. Efficient simulation with a universal quantum computer

Universal quantum computation



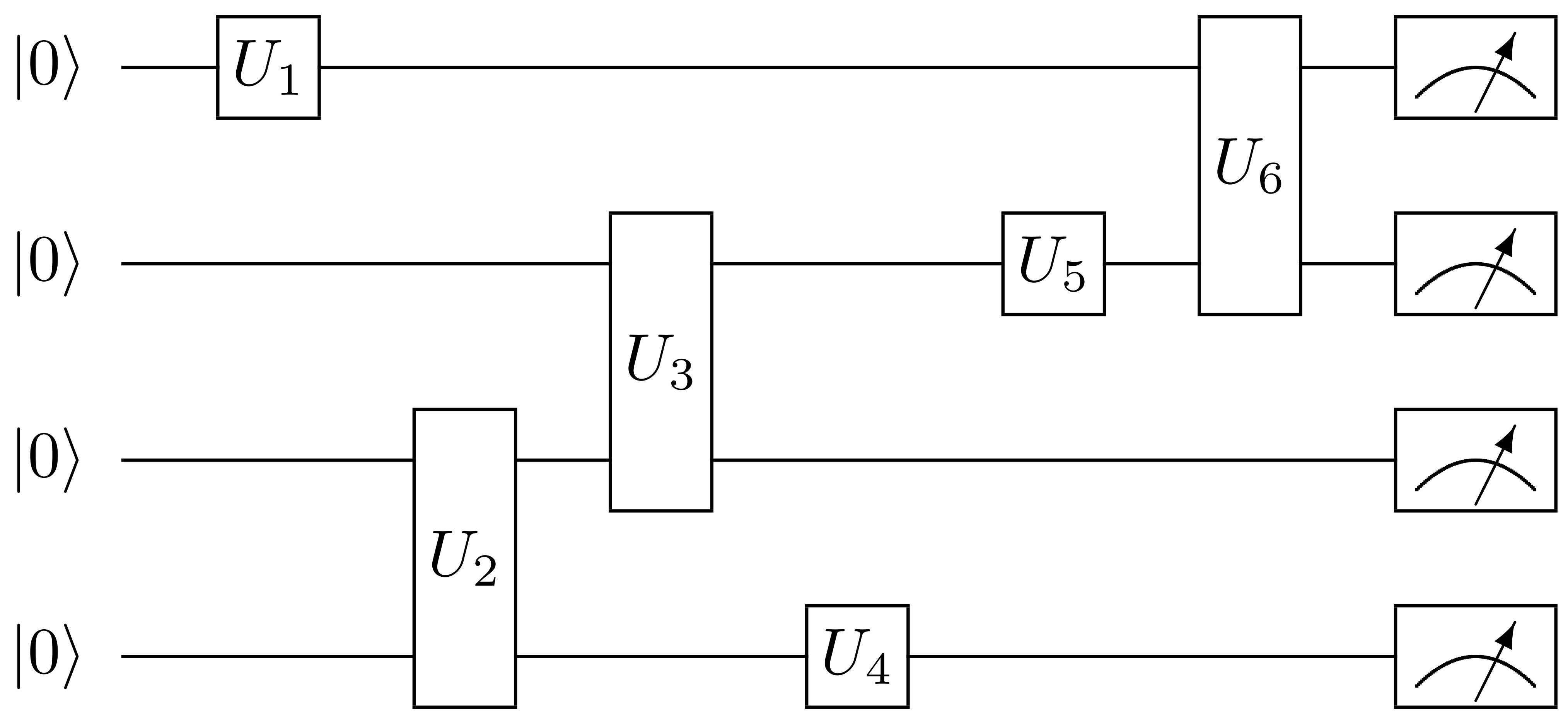
The simulation problem

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

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

1. Efficient simulation with a universal quantum computer

Universal quantum computation



The simulation problem

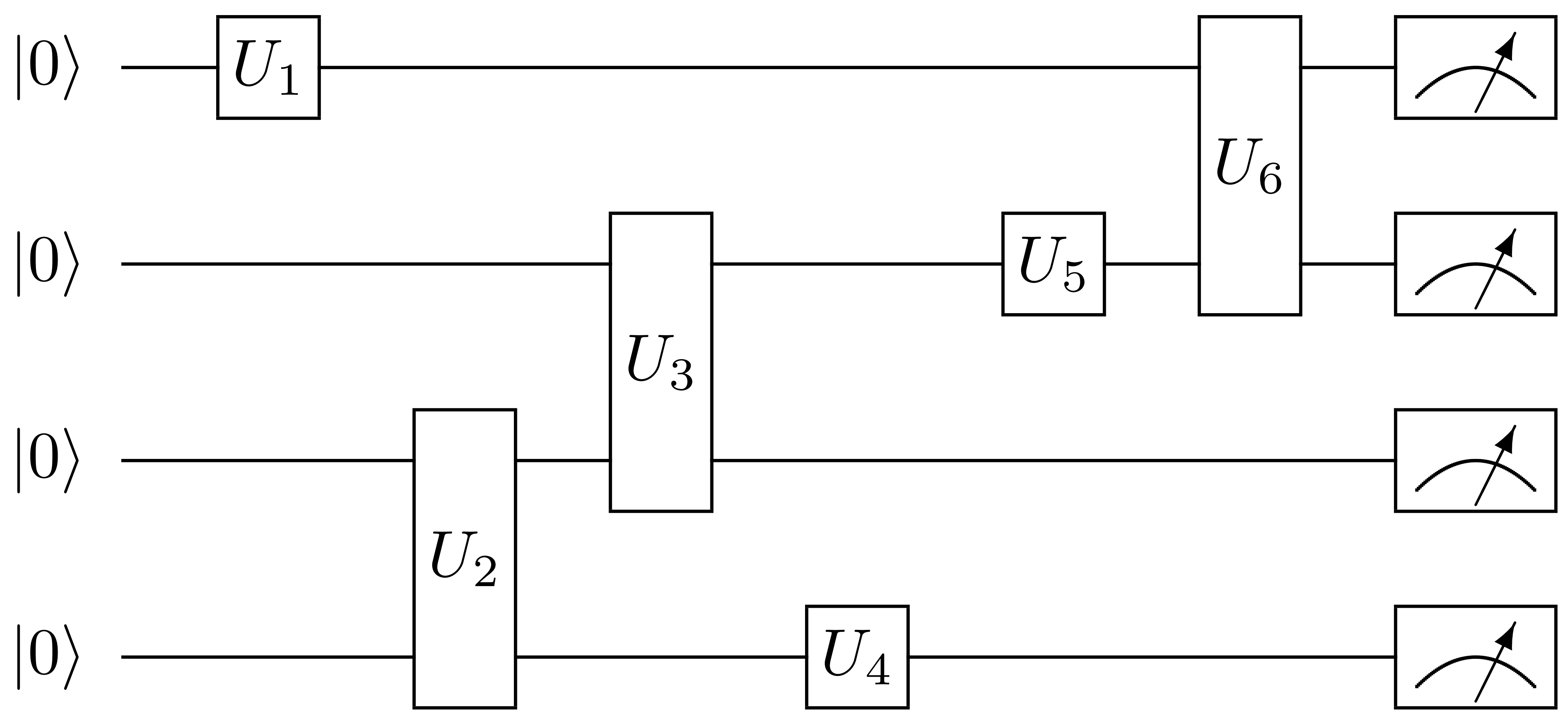
The dynamics of a quantum system are determined by its *Hamiltonian* H.

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

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).

1. Efficient simulation with a universal quantum computer

Universal quantum computation



The simulation problem

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

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

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).

This is as hard as anything a quantum computer can do!

1. Efficient simulation with a universal quantum computer

[Lloyd 96; Berry, Ahokas, Cleve, Sanders 05]

1. Efficient simulation with a universal quantum computer

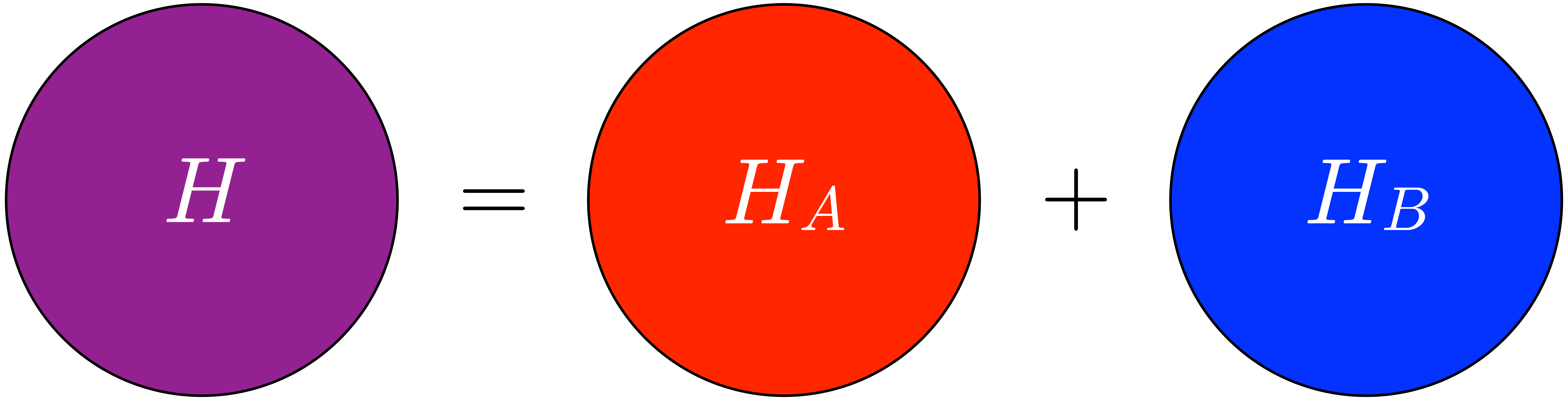
Suppose $H = \sum_j H_j$ where each H_j is individually easy to simulate

[Lloyd 96; Berry, Ahokas, Cleve, Sanders 05]

1. Efficient simulation with a universal quantum computer

Suppose $H = \sum_j H_j$ where each H_j is individually easy to simulate

Main idea:

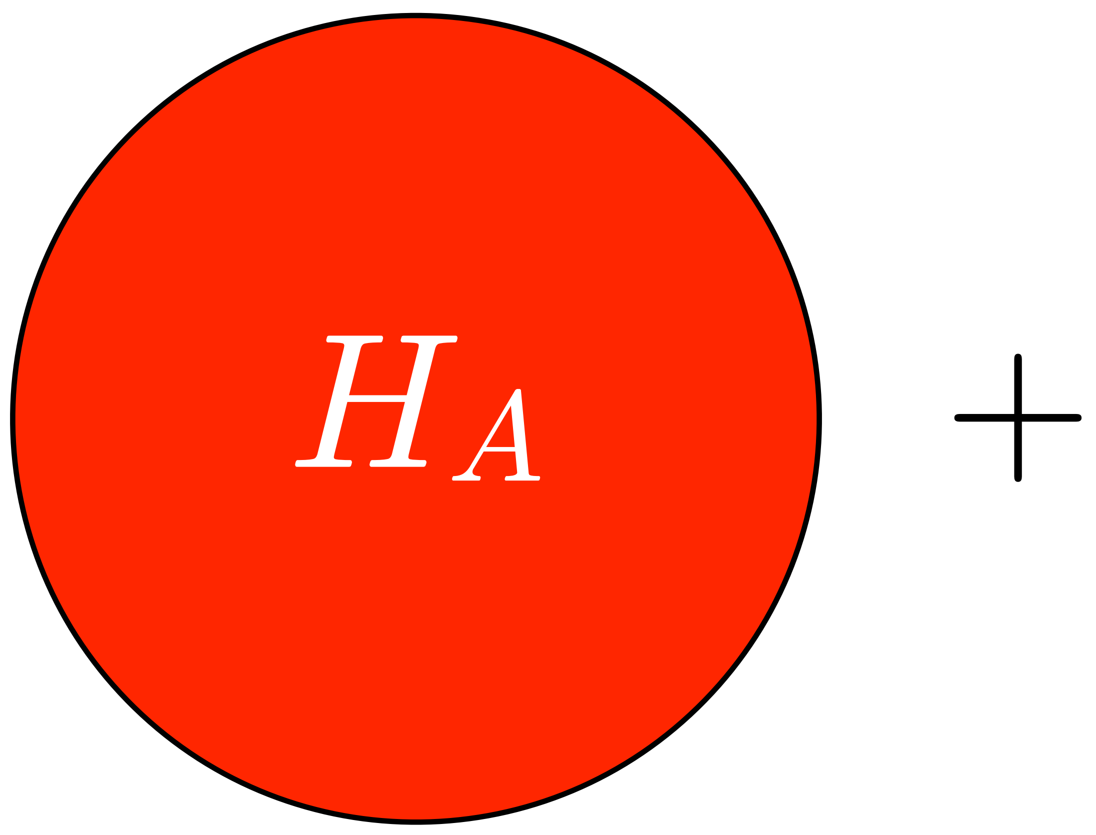


[Lloyd 96; Berry, Ahokas, Cleve, Sanders 05]

1. Efficient simulation with a universal quantum computer

Suppose $H = \sum_j H_j$ where each H_j is individually easy to simulate

Main idea:

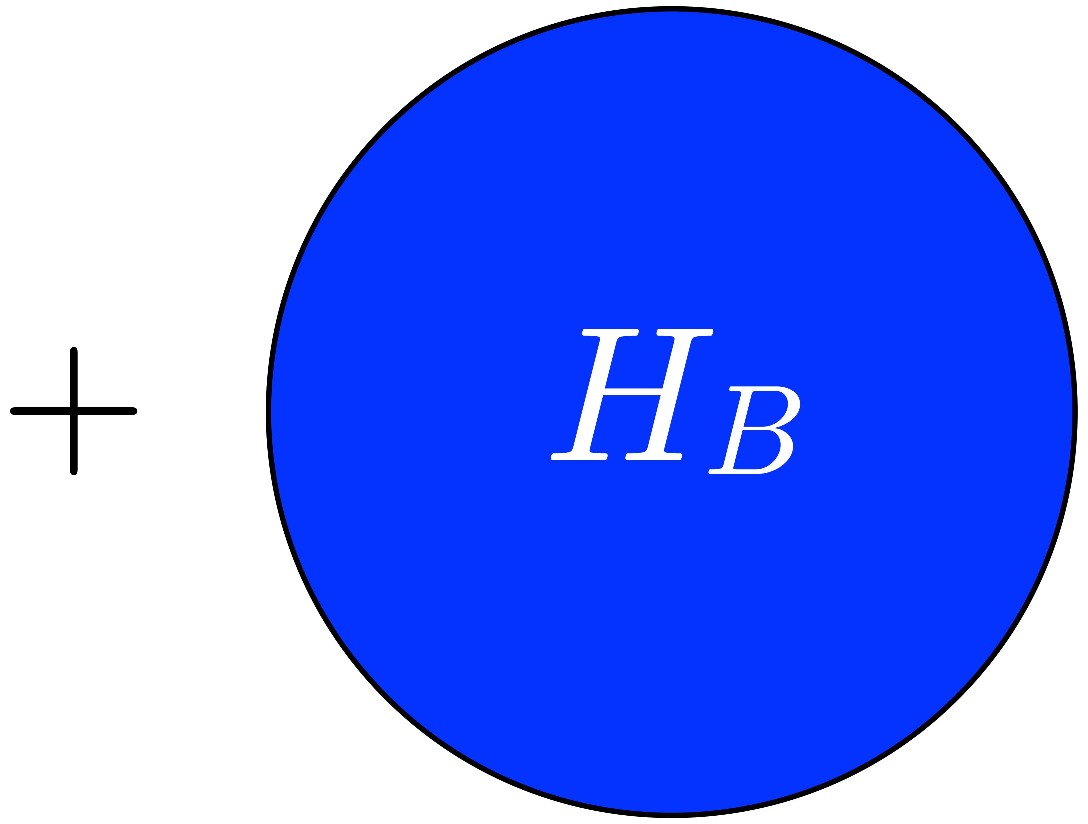


[Lloyd 96; Berry, Ahokas, Cleve, Sanders 05]

1. Efficient simulation with a universal quantum computer

Suppose $H = \sum_j H_j$ where each H_j is individually easy to simulate

Main idea:

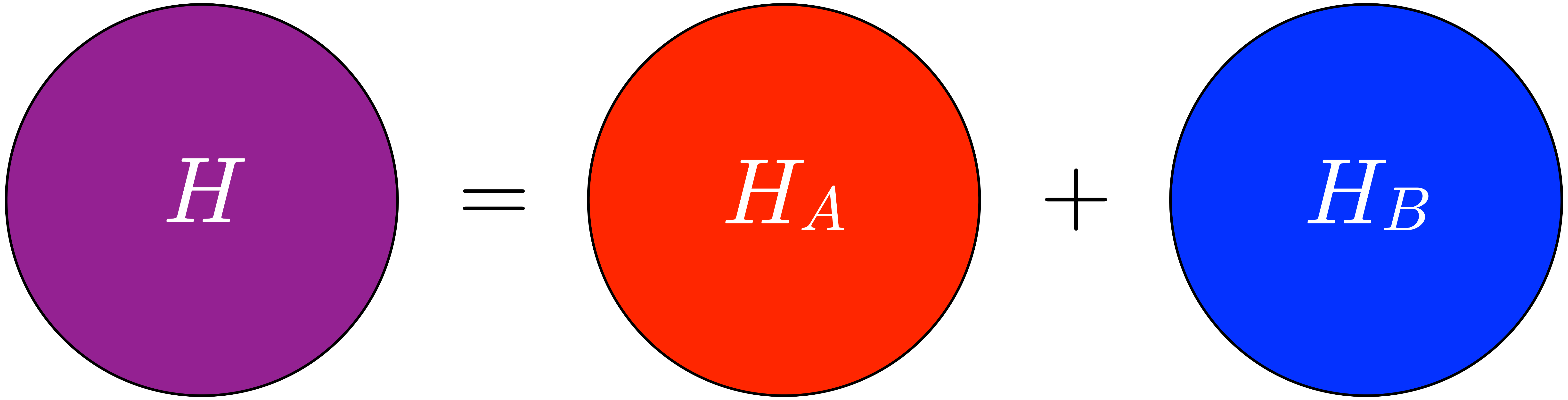


[Lloyd 96; Berry, Ahokas, Cleve, Sanders 05]

1. Efficient simulation with a universal quantum computer

Suppose $H = \sum_j H_j$ where each H_j is individually easy to simulate

Main idea:

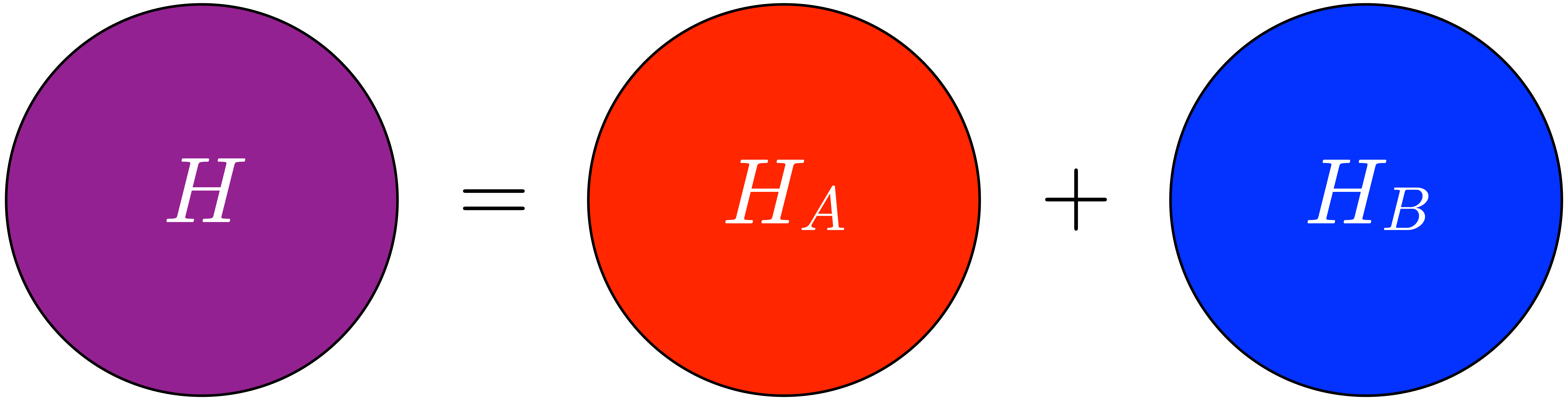


[Lloyd 96; Berry, Ahokas, Cleve, Sanders 05]

1. Efficient simulation with a universal quantum computer

Suppose $H = \sum_j H_j$ where each H_j is individually easy to simulate

Main idea:



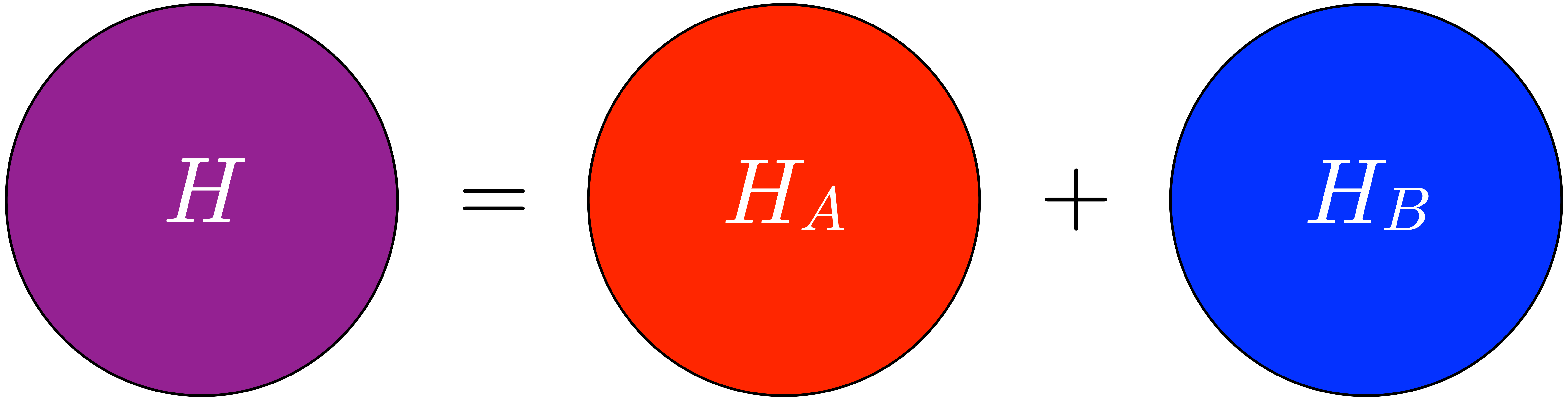
Complexity: $O(t^2/\epsilon)$

[Lloyd 96; Berry, Ahokas, Cleve, Sanders 05]

1. Efficient simulation with a universal quantum computer

Suppose $H = \sum_j H_j$ where each H_j is individually easy to simulate

Main idea:



Complexity: $O(t^2/\epsilon)$

Higher-order version (order k): $O\big(5^k t (t/\epsilon)^{1/k}\big)$

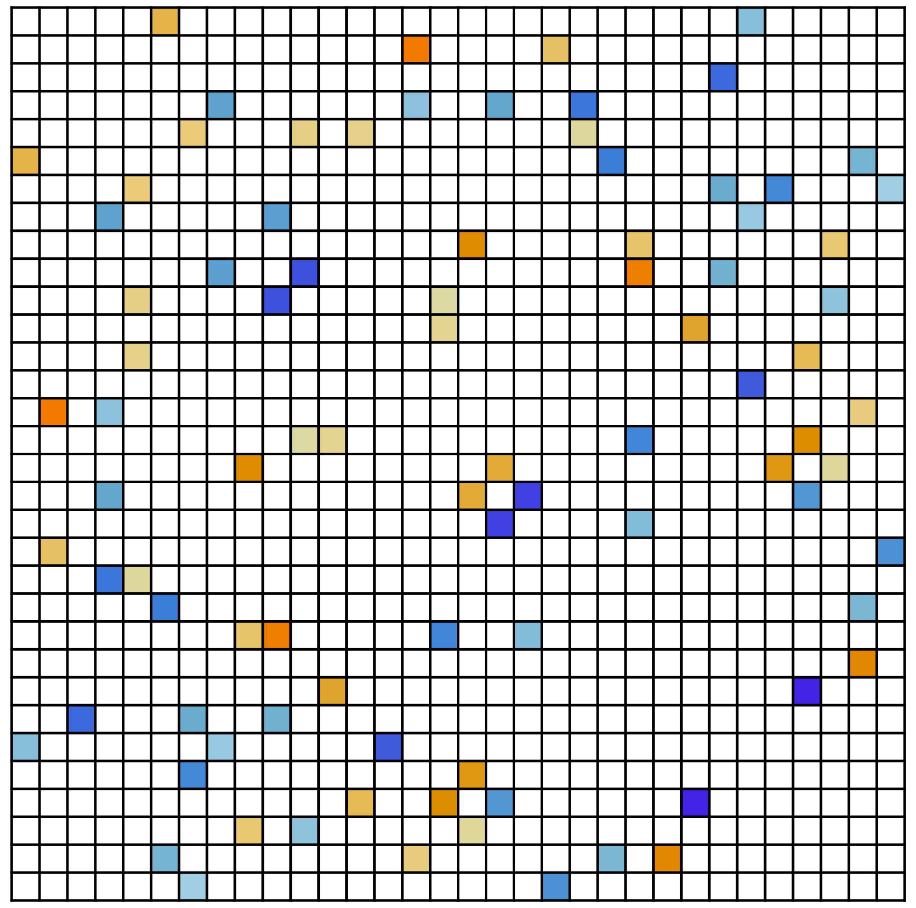
[Lloyd 96; Berry, Ahokas, Cleve, Sanders 05]

2. Simulating sparse Hamiltonians

[Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 03; Aharonov, Ta-Shma 03]

2. Simulating sparse Hamiltonians

Sparse Hamiltonians [Aharonov, Ta-Shma 03]

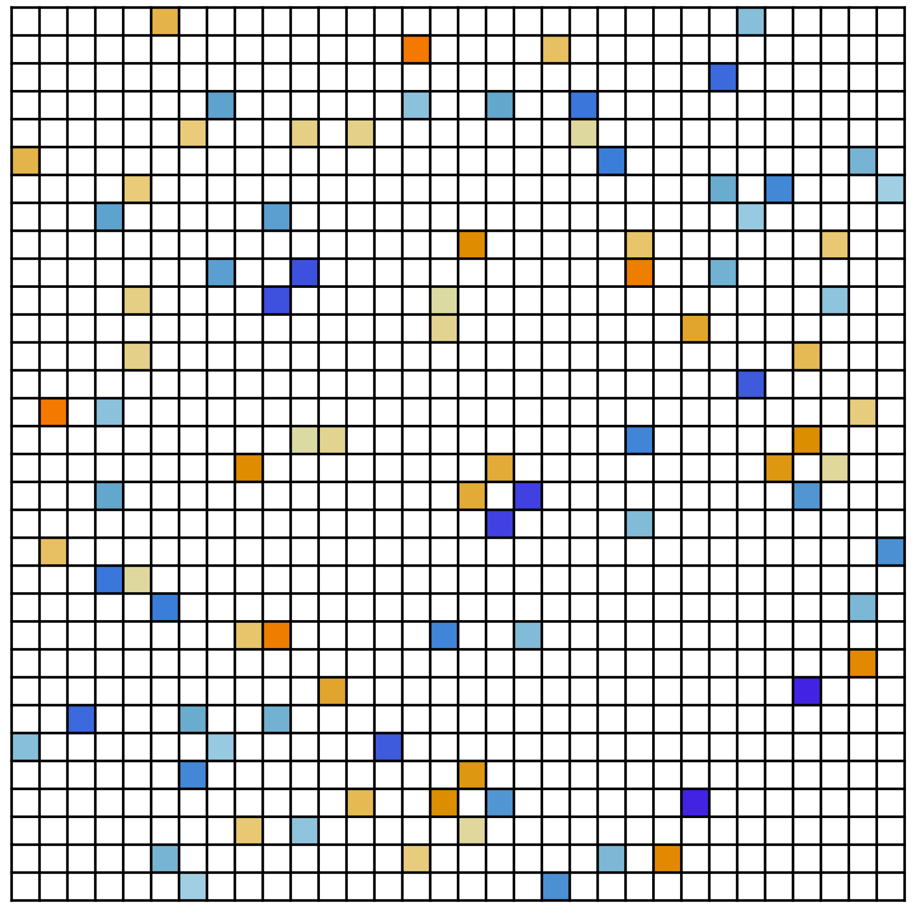
$H =$ 

[Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 03; Aharonov, Ta-Shma 03]

2. Simulating sparse Hamiltonians

Sparse Hamiltonians [Aharonov, Ta-Shma 03]

For any given row, the locations of the nonzero entries and their values can be computed efficiently

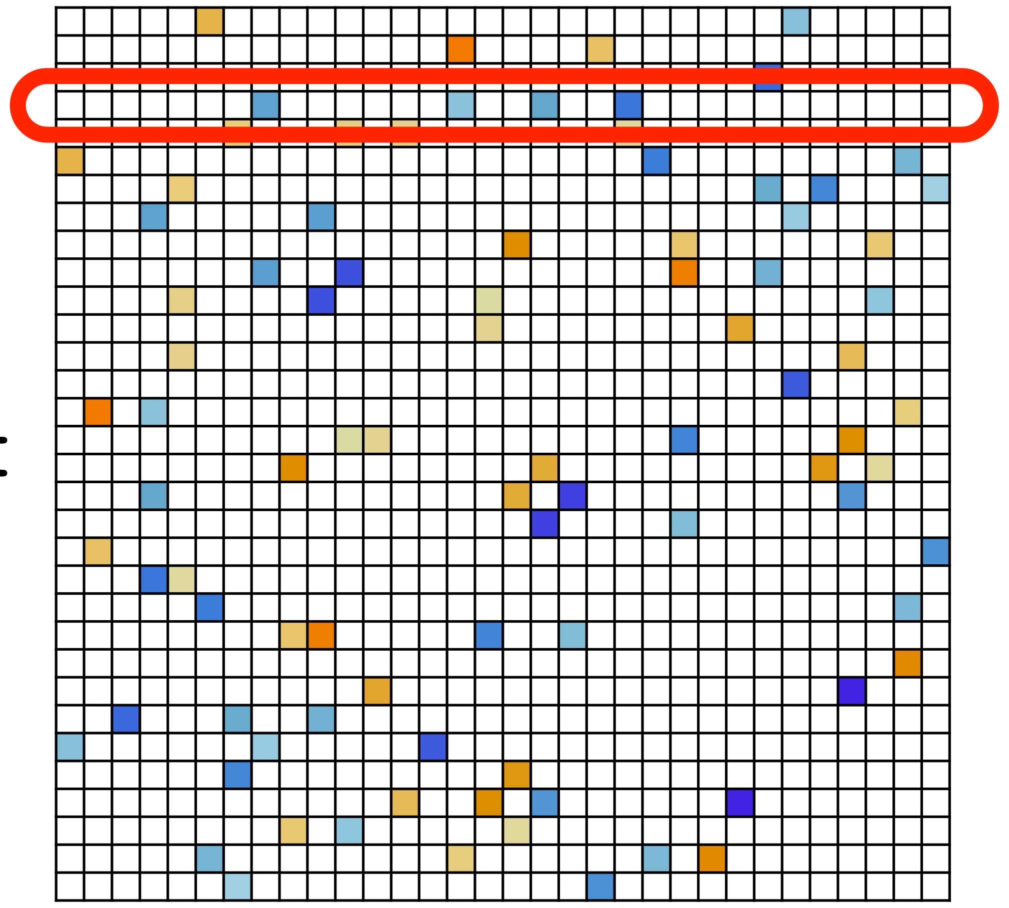
$H =$ 

[Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 03; Aharonov, Ta-Shma 03]

2. Simulating sparse Hamiltonians

Sparse Hamiltonians [Aharonov, Ta-Shma 03]

For any given row, the locations of the nonzero entries and their values can be computed efficiently

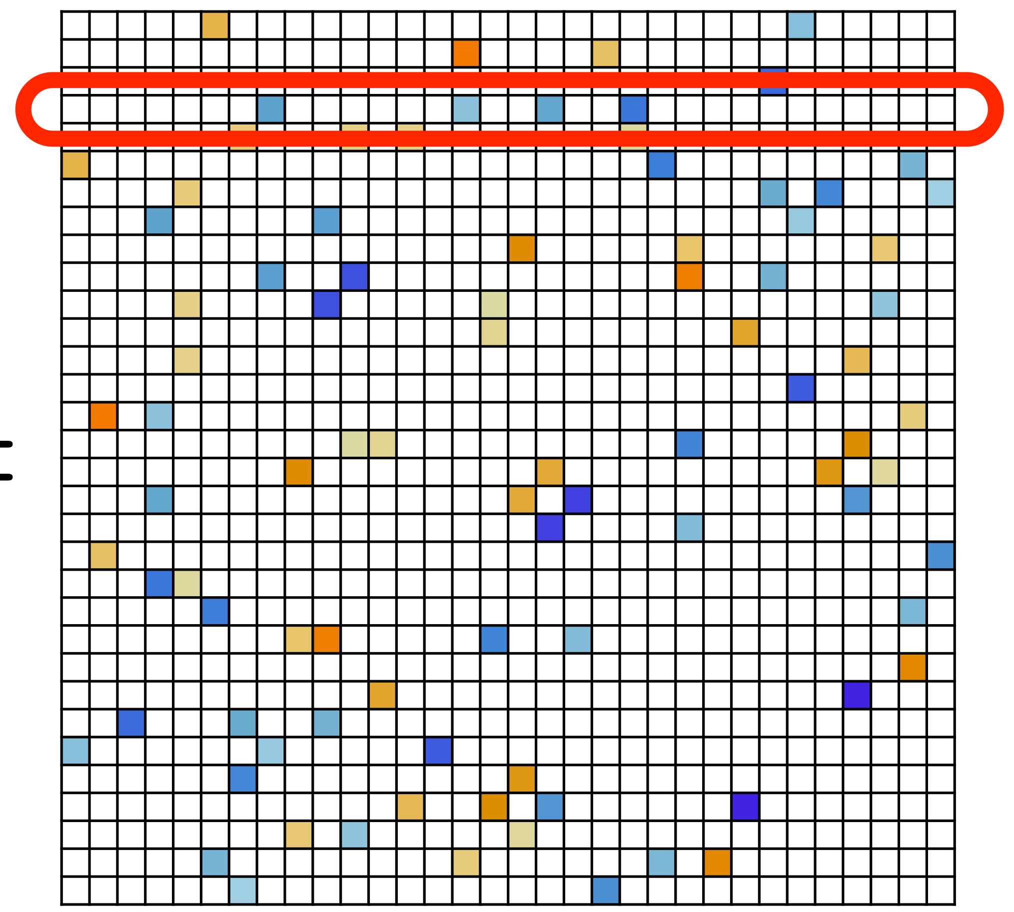
$H =$ 

[Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 03; Aharonov, Ta-Shma 03]

2. Simulating sparse Hamiltonians

Sparse Hamiltonians [Aharonov, Ta-Shma 03]

For any given row, the locations of the nonzero entries and their values can be computed efficiently

$H =$ 

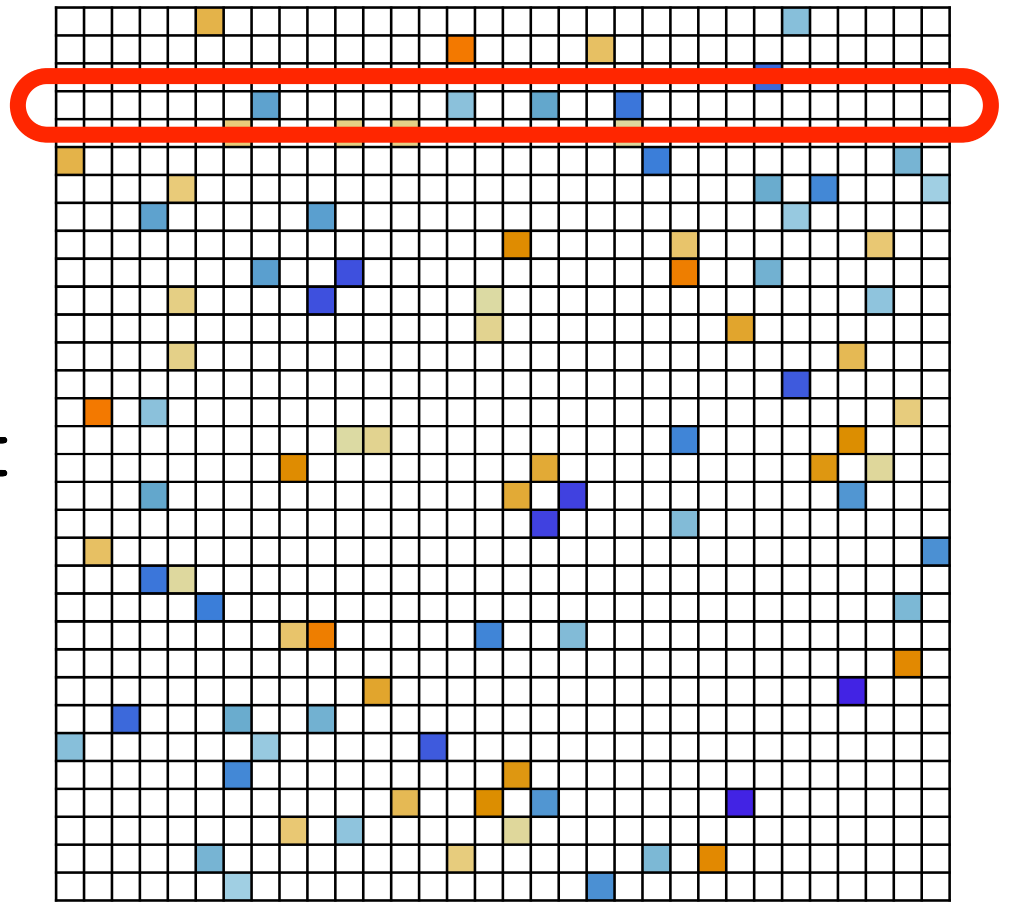
Main idea: Color the edges of the graph of H. Then the simulation breaks into small pieces that are easy to handle.

[Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 03; Aharonov, Ta-Shma 03]

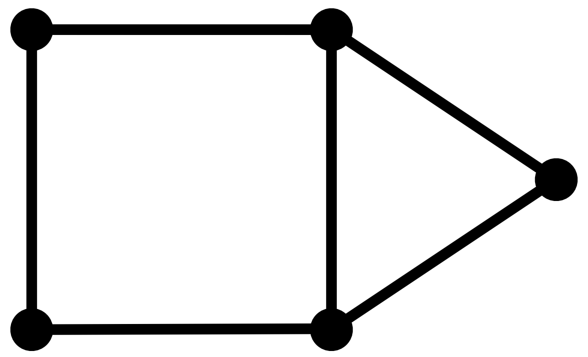
2. Simulating sparse Hamiltonians

Sparse Hamiltonians [Aharonov, Ta-Shma 03]

For any given row, the locations of the nonzero entries and their values can be computed efficiently

$H =$ 

Main idea: Color the edges of the graph of H. Then the simulation breaks into small pieces that are easy to handle.

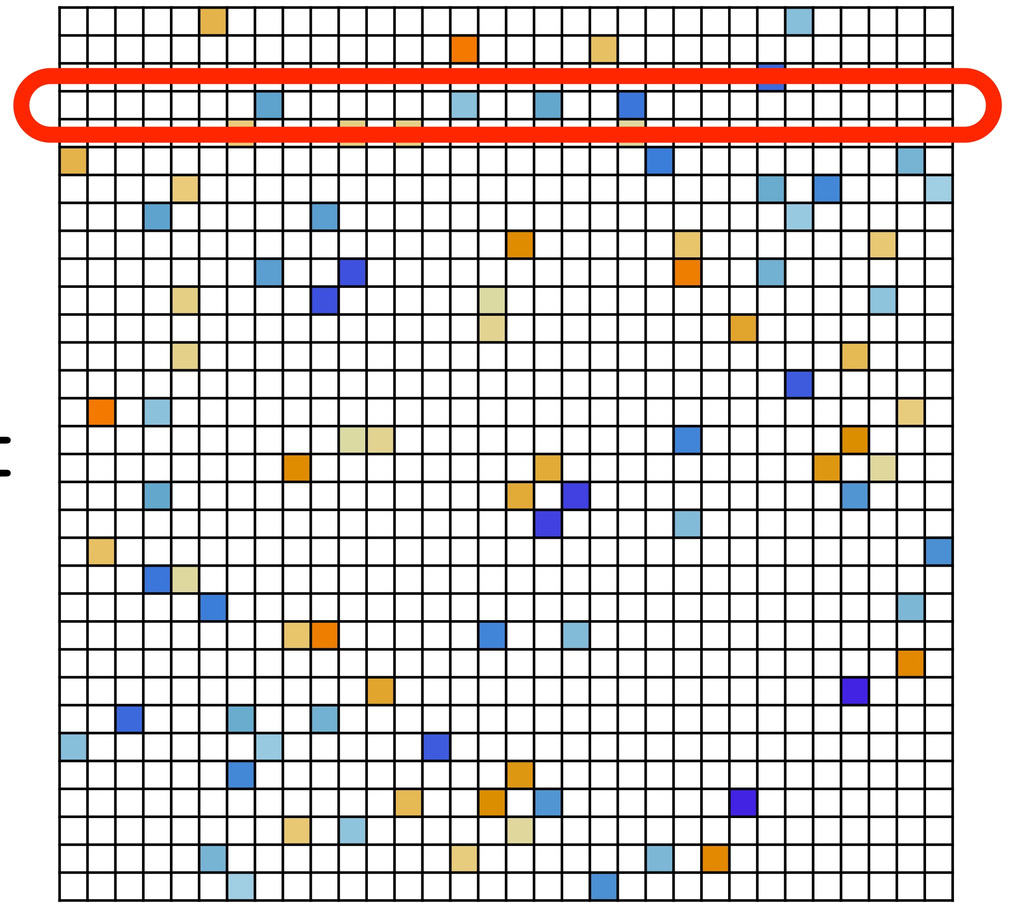


[Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 03; Aharonov, Ta-Shma 03]

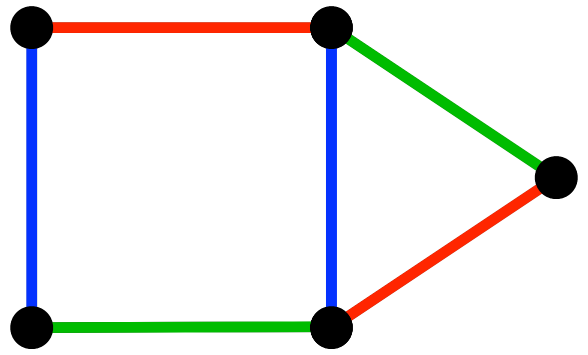
2. Simulating sparse Hamiltonians

Sparse Hamiltonians [Aharonov, Ta-Shma 03]

For any given row, the locations of the nonzero entries and their values can be computed efficiently

$H =$ 

Main idea: Color the edges of the graph of H. Then the simulation breaks into small pieces that are easy to handle.

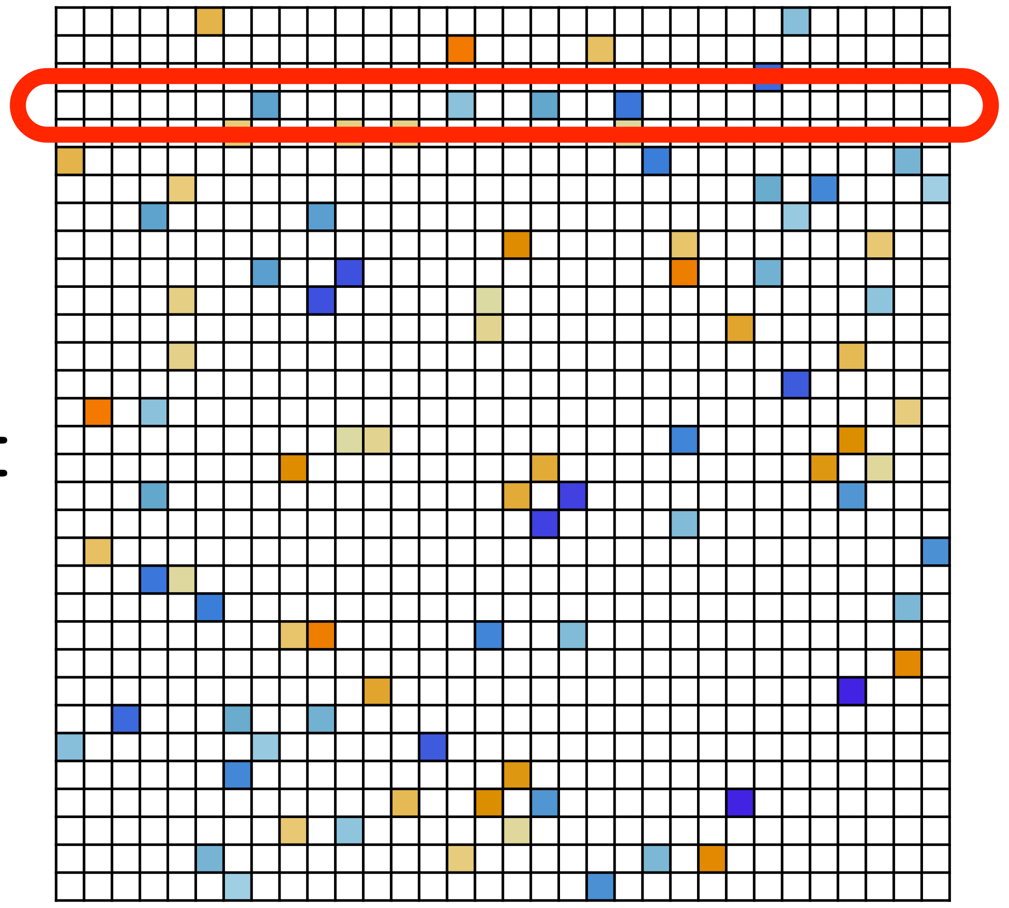


[Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 03; Aharonov, Ta-Shma 03]

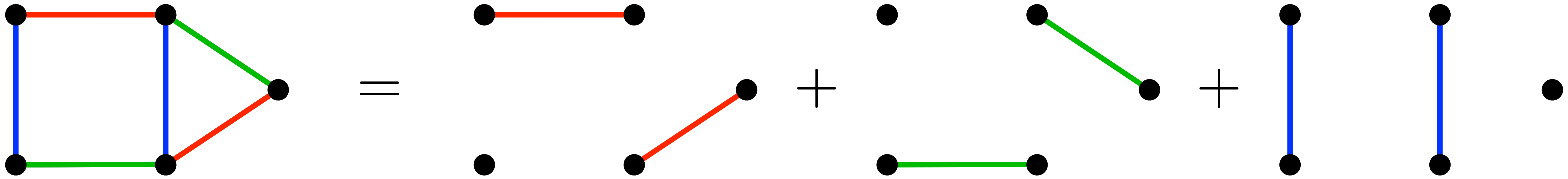
2. Simulating sparse Hamiltonians

Sparse Hamiltonians [Aharonov, Ta-Shma 03]

For any given row, the locations of the nonzero entries and their values can be computed efficiently

$H =$ 

Main idea: Color the edges of the graph of H. Then the simulation breaks into small pieces that are easy to handle.



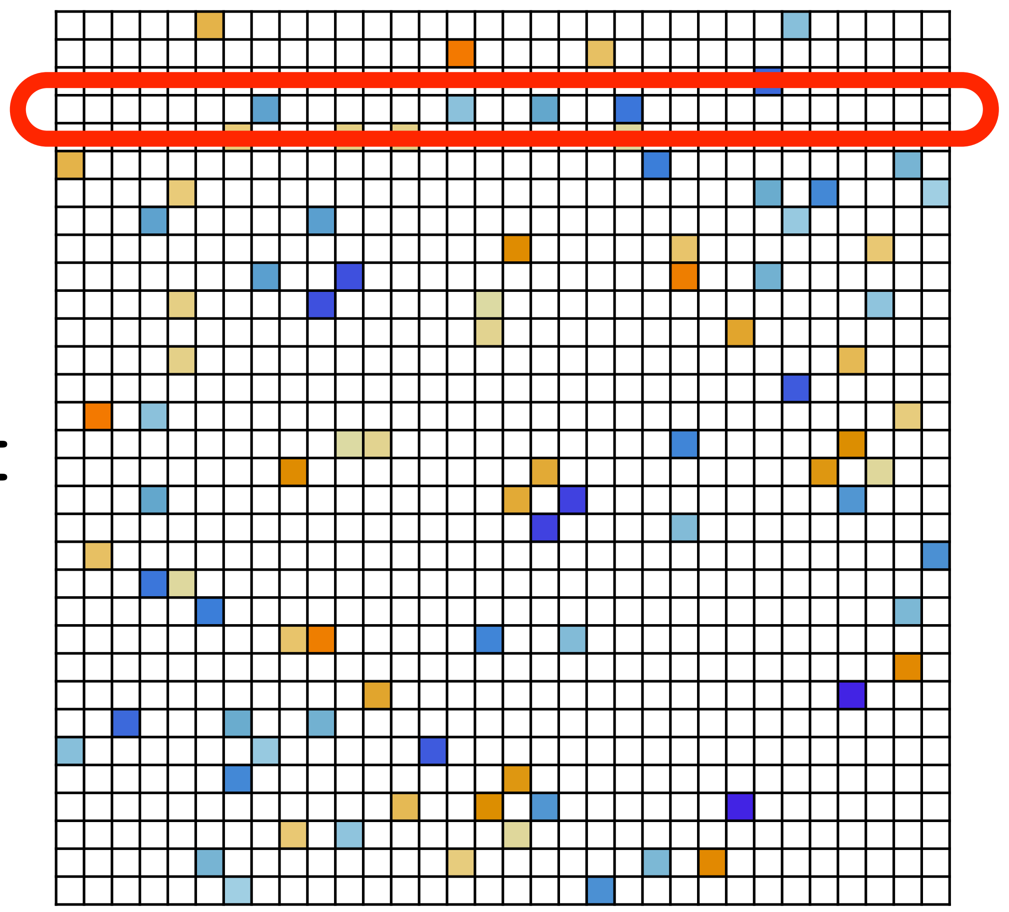
[Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 03; Aharonov, Ta-Shma 03]

2. Simulating sparse Hamiltonians

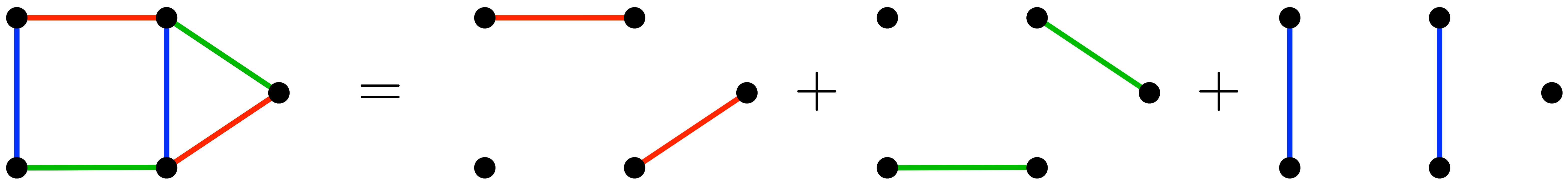
Sparse Hamiltonians [Aharonov, Ta-Shma 03]

For any given row, the locations of the nonzero entries and their values can be computed efficiently

$$H =$$



Main idea: Color the edges of the graph of H. Then the simulation breaks into small pieces that are easy to handle.



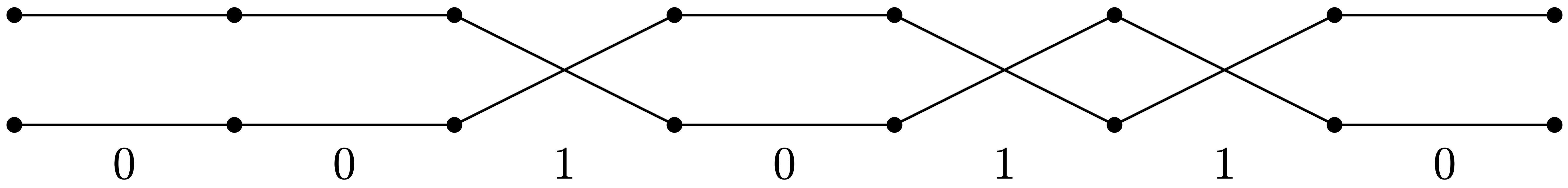
A sparse graph can be efficiently colored using only local information, so this gives efficient simulations.

[Childs, Cleve, Deotto, Farhi, Gutmann, Spielman 03; Aharonov, Ta-Shma 03]

3. Simulating quantum mechanics in real time

3. Simulating quantum mechanics in real time

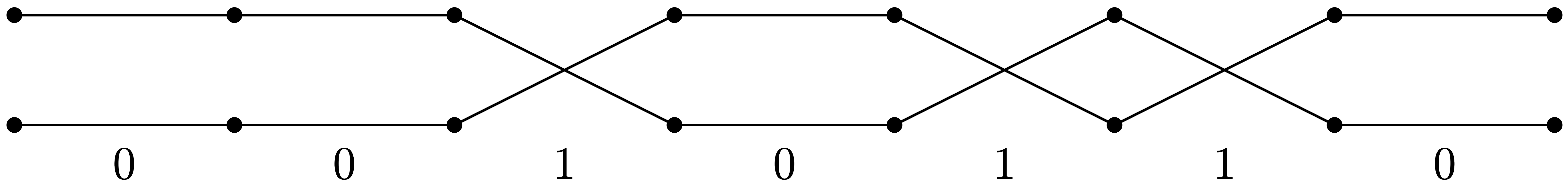
No fast-forwarding theorem: Simulating Hamiltonian dynamics for time t requires $\Omega(t)$ gates.



[Berry, Ahokas, Cleve, Sanders 05]

3. Simulating quantum mechanics in real time

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

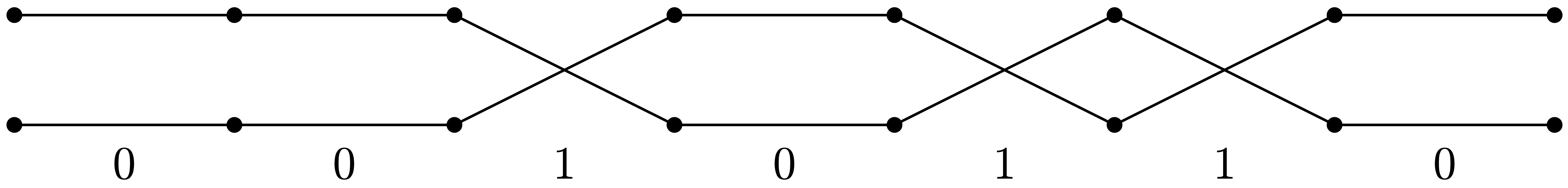


[Berry, Ahokas, Cleve, Sanders 05]

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

3. Simulating quantum mechanics in real time

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



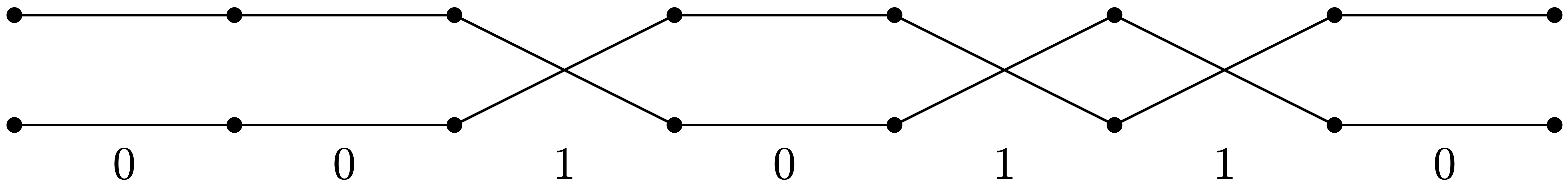
[Berry, Ahokas, Cleve, Sanders 05]

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

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

3. 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 $O(t)$?

Systems simulate their own dynamics in real time!

3. Simulating quantum mechanics in real time

[Childs 10; Berry, Childs 12]

3. Simulating quantum mechanics in real time

Challenge: mismatch between

- Schrödinger dynamics (continuous time) and
- the quantum circuit model (discrete time)

[Childs 10; Berry, Childs 12]

3. Simulating quantum mechanics in real time

Challenge: mismatch between

- Schrödinger dynamics (continuous time) and
- the quantum circuit model (discrete time)

Main idea: introduce a *discrete-time quantum walk* corresponding to the Hamiltonian

- discrete dynamics
- easy to implement
- efficiently carries spectral information about the Hamiltonian

[Childs 10; Berry, Childs 12]

3. Simulating quantum mechanics in real time

Challenge: mismatch between

- Schrödinger dynamics (continuous time) and
- the quantum circuit model (discrete time)

Main idea: introduce a *discrete-time quantum walk* corresponding to the Hamiltonian

- discrete dynamics
- easy to implement
- efficiently carries spectral information about the Hamiltonian

Complexity: $O(t/\sqrt{\epsilon})$

linear in t!

[Childs 10; Berry, Childs 12]

4. High-precision simulation?

Can we improve the dependence on ϵ?

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

4. High-precision simulation?

Can we improve the dependence on ϵ?

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

- computing π
- boosting a bounded-error subroutine
- Solovay-Kitaev circuit synthesis
- and more…

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

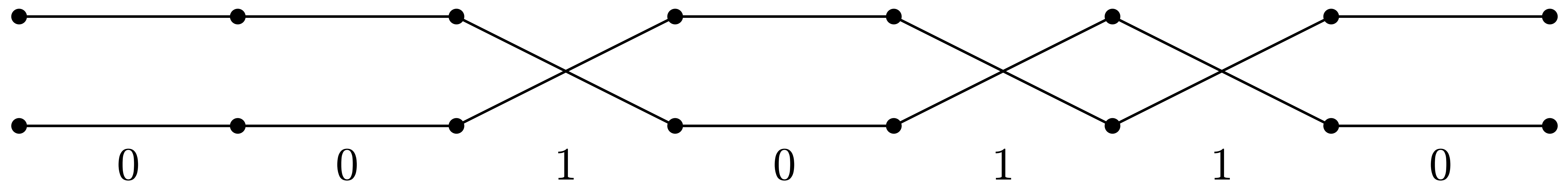
4. High-precision simulation?

Can we improve the dependence on ϵ?

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

- computing π
- boosting a bounded-error subroutine
- Solovay-Kitaev circuit synthesis
- and more…

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



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

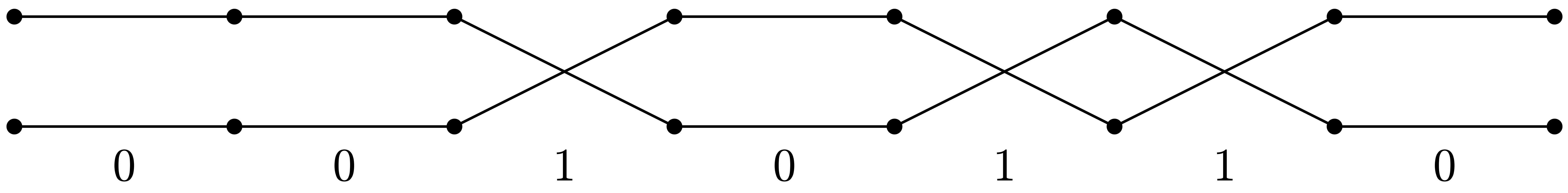
4. High-precision simulation?

Can we improve the dependence on ϵ?

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

- computing 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: $O(1/\sqrt{\epsilon})$

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

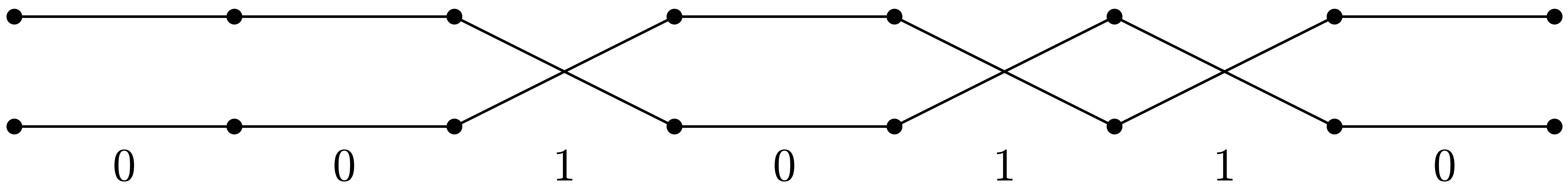
4. High-precision simulation?

Can we improve the dependence on ϵ?

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

- computing 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 (kth order): $O\left(5^k/\epsilon^{1/k}\right)$

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

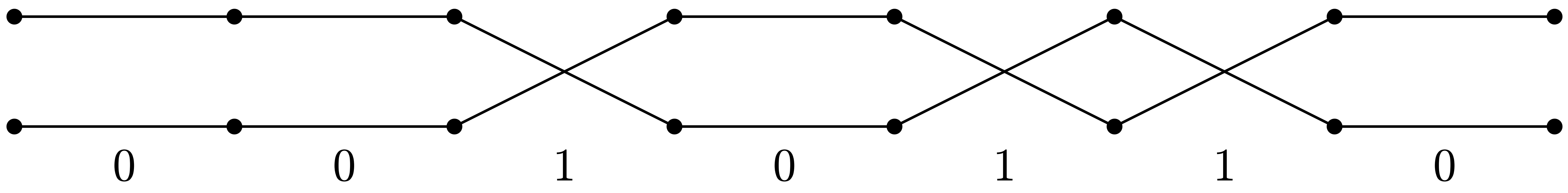
4. High-precision simulation?

Can we improve the dependence on ϵ?

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

- computing π
- boosting a bounded-error subroutine
- Solovay-Kitaev circuit synthesis
- and more…

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



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

Product formulas (kth order): $O\big(5^k/\epsilon^{1/k}\big)$

Can we do better?

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

4. High-precision simulation

Yes! There is a simulation with complexity $O\big(t \frac{\log(t/\epsilon)}{\log\log(t/\epsilon)}\big)$.

exponential improvement in ϵ!

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

4. High-precision simulation

Yes! There is a simulation with complexity $O\big(t\frac{\log(t/\epsilon)}{\log\log(t/\epsilon)}\big)$.

exponential improvement in ϵ!

Main idea:

- Consider the truncated Taylor series $e^{-iHt} = \sum_{k=0}^{\infty} \frac{(-iHt)^k}{k!} \approx \sum_{k=K}^{\infty} \frac{(-iHt)^k}{k!}$
- Expand H as a linear combination of unitary operators
- Directly implement the overall linear combination by *oblivious amplitude amplification*

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

5. An optimal tradeoff

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

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

[Low, Chuang 16]

5. An optimal tradeoff

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

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

Recent work, using an alternative method for implementing a linear combination of quantum walk steps, achieves the lower bound.

optimal tradeoff between t and ϵ!

[Low, Chuang 16]

5. An optimal tradeoff

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

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

Recent work, using an alternative method for implementing a linear combination of quantum walk steps, achieves the lower bound.

optimal tradeoff between t and ϵ!

Main idea:

- Encode the eigenvalues of H in a two-dimensional subspace
- Manipulate those eigenvalues using a carefully-chosen sequence of single-qubit rotations (inspired by quantum control technique of *composite pulses*)

[Low, Chuang 16]

6. Real-time simulation revisited

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

[Haah, Hastings, Kothari, Low 18]

6. Real-time simulation revisited

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. To simulate for constant time, best previous methods give:

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

[Haah, Hastings, Kothari, Low 18]

6. Real-time simulation revisited

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. To simulate for constant time, best previous methods 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!

[Haah, Hastings, Kothari, Low 18]

6. Real-time simulation revisited

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. To simulate for constant time, best previous methods 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 $O(n)$ gates, $O(1)$ depth

optimal n dependence!

[Haah, Hastings, Kothari, Low 18]

6. Real-time simulation revisited

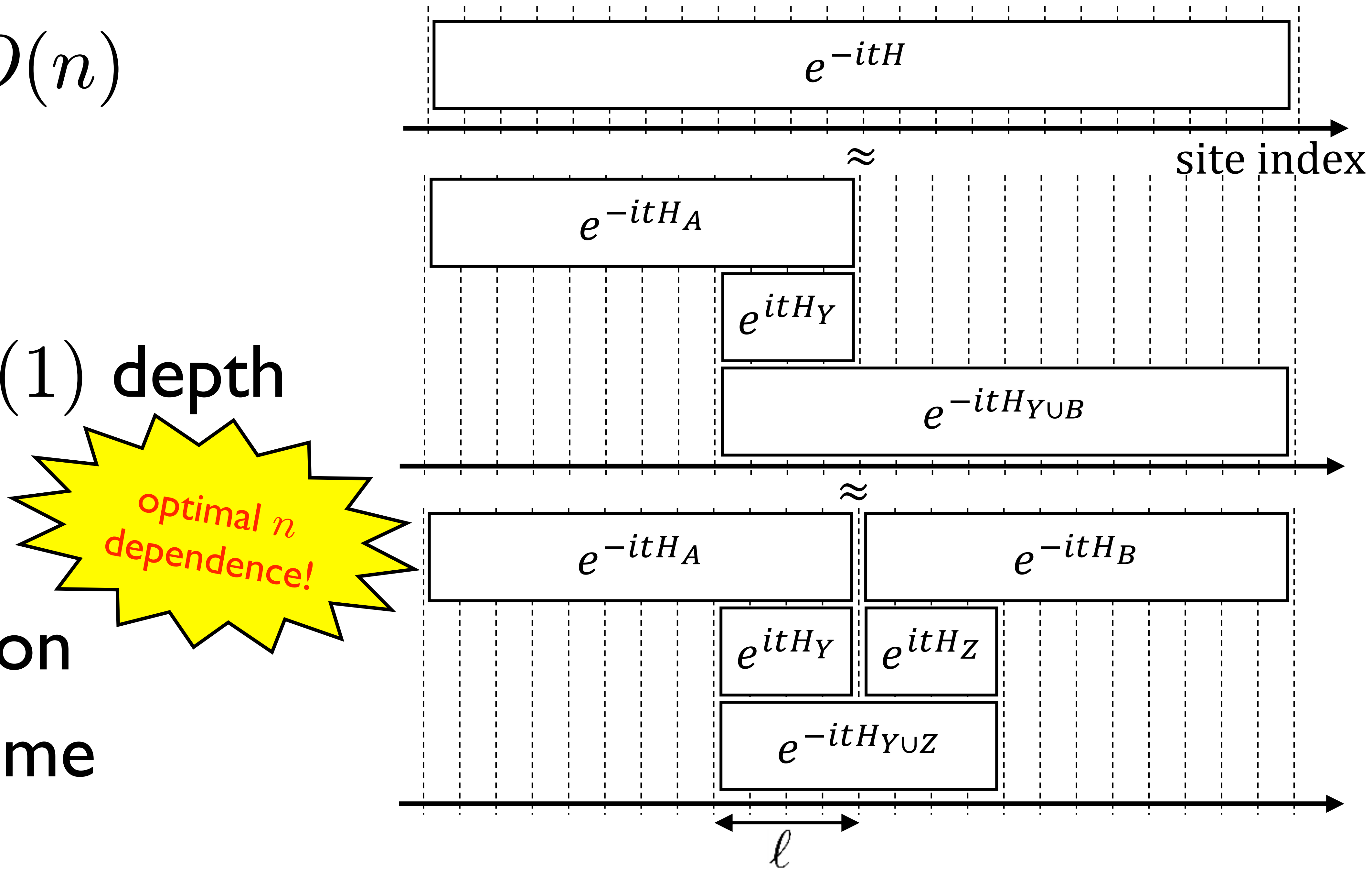
We've focused on the complexity as a function of t (evolution time) and ϵ (precision).
What about the dependence on system size?

Consider a system of n spins with nearest-neighbor interactions. To simulate for constant time, best previous methods 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 $O(n)$ gates, $O(1)$ depth

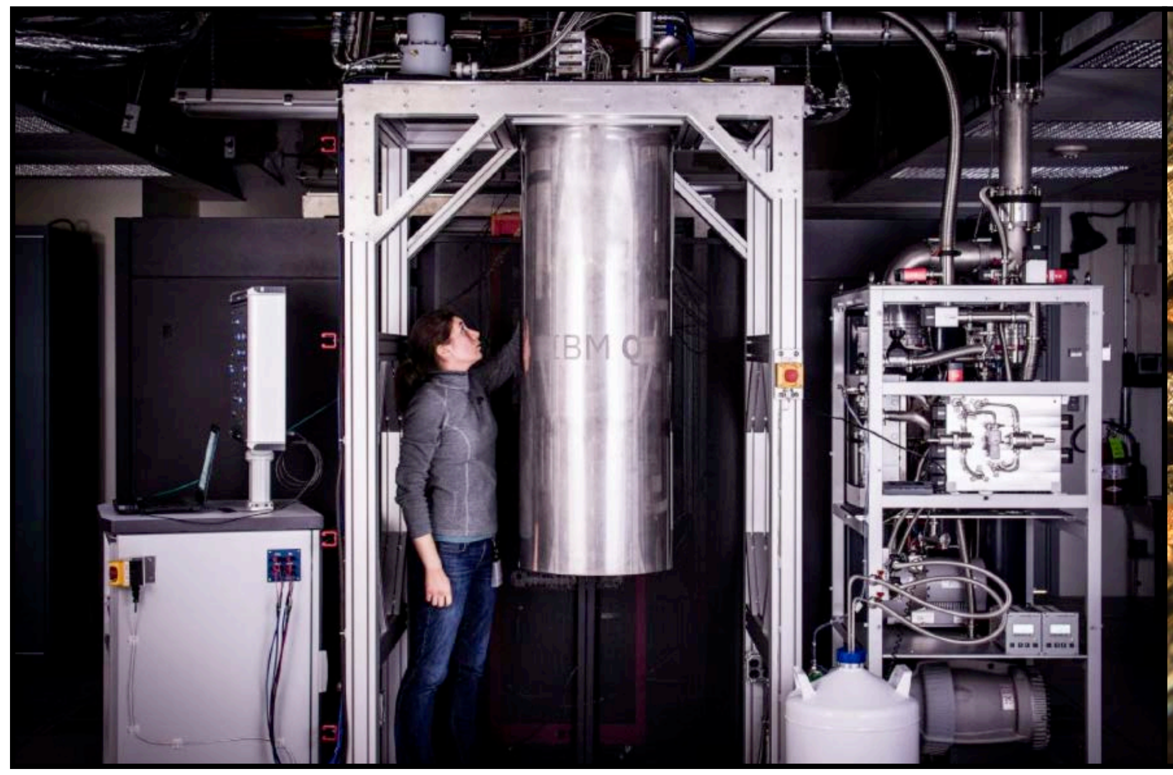


Main ideas:

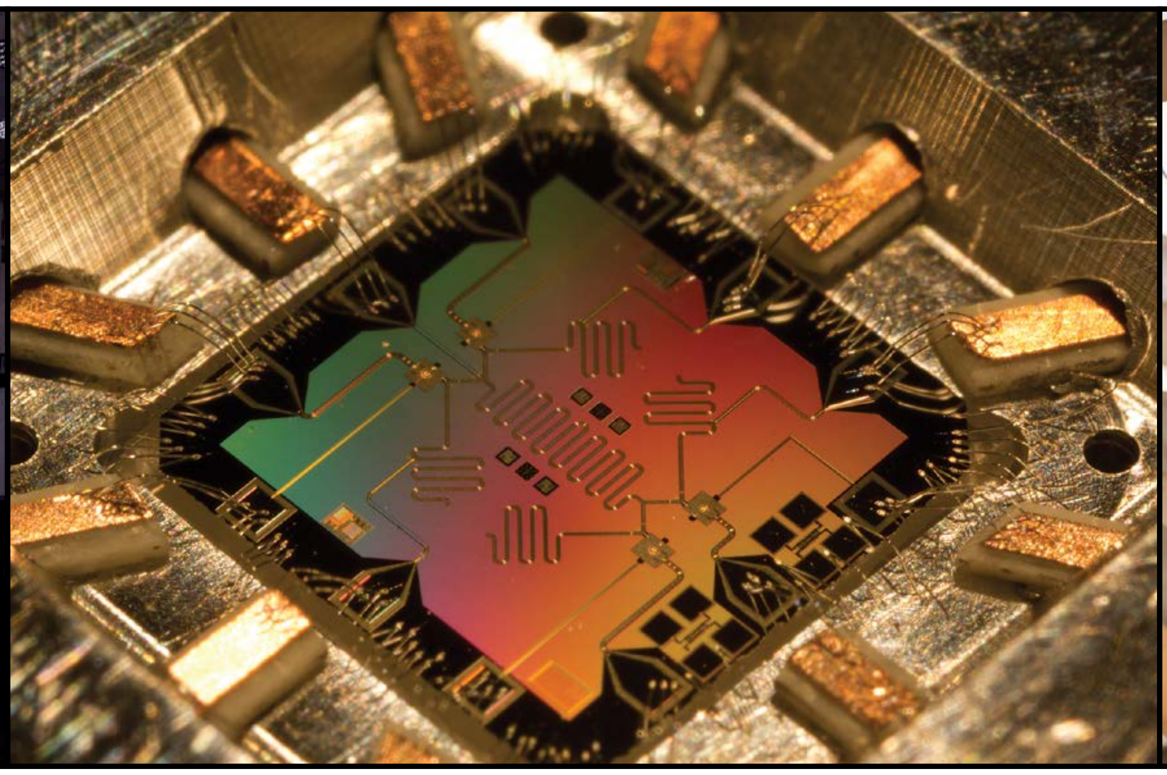
- Lieb-Robinson bound limits the speed of propagation
- Simulate small, overlapping regions, with negative-time evolutions to correct the boundaries

[Haah, Hastings, Kothari, Low 18]

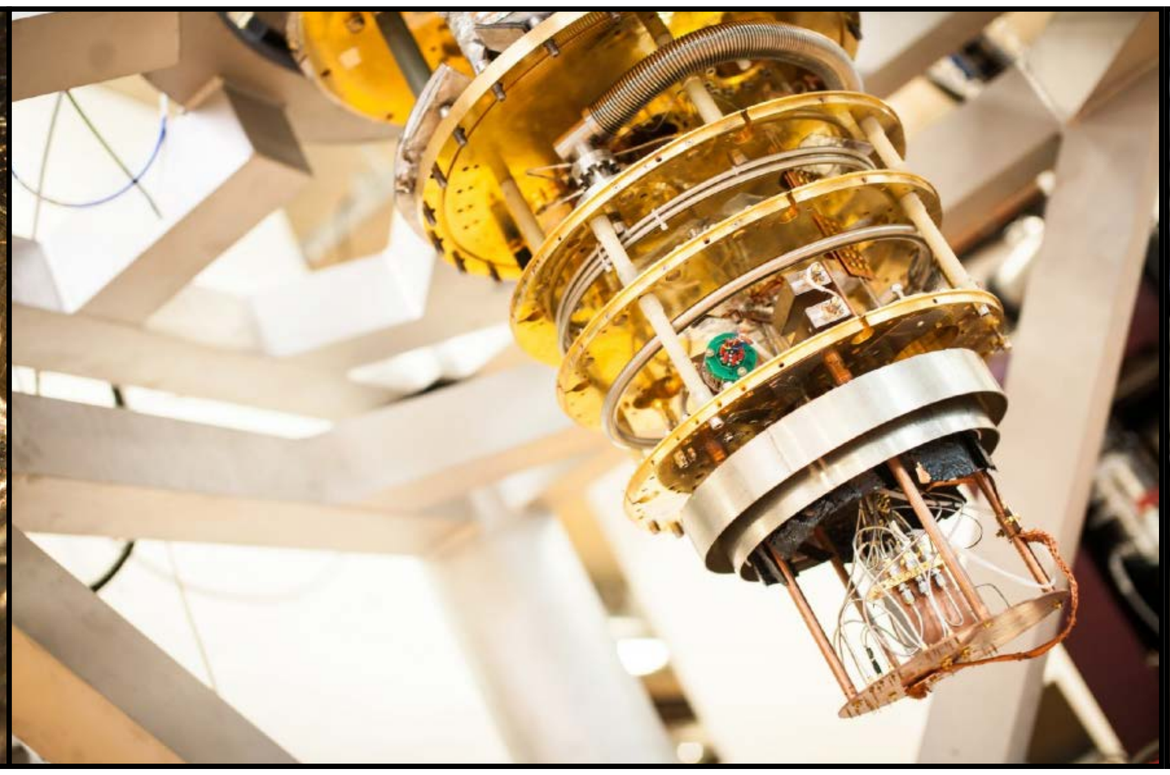
7. Making quantum simulation practical



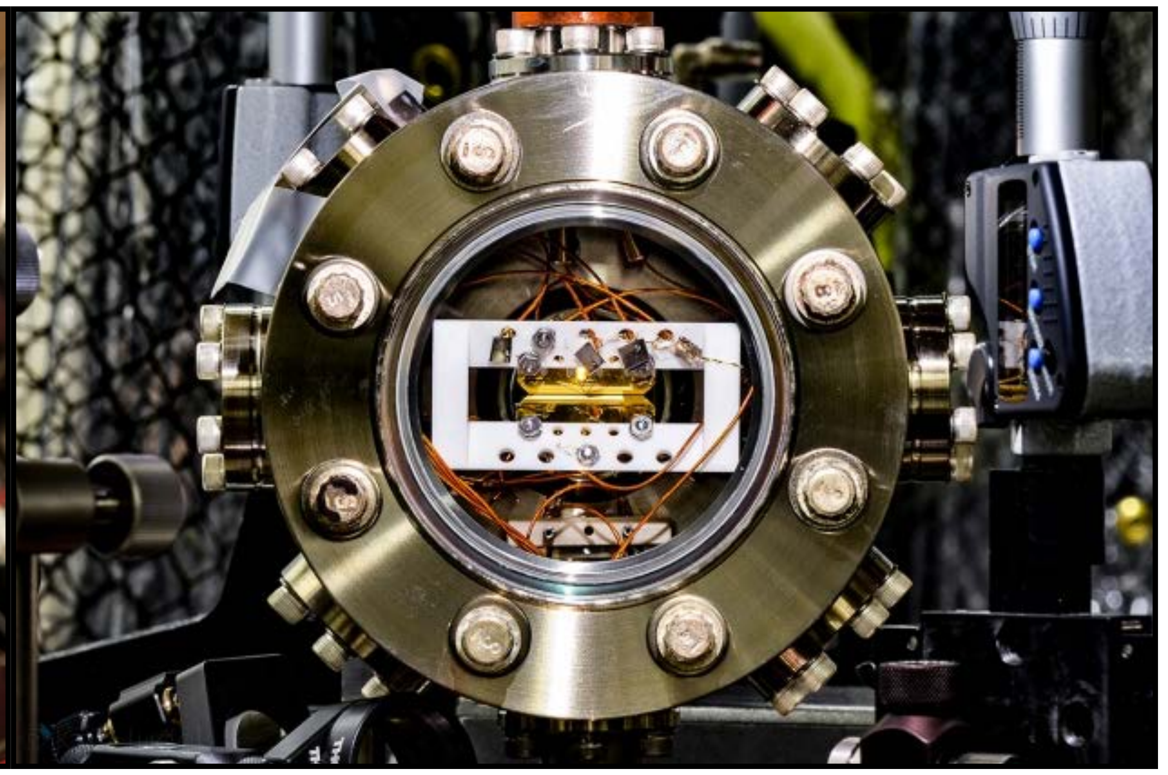
IBM



Google/UCSB

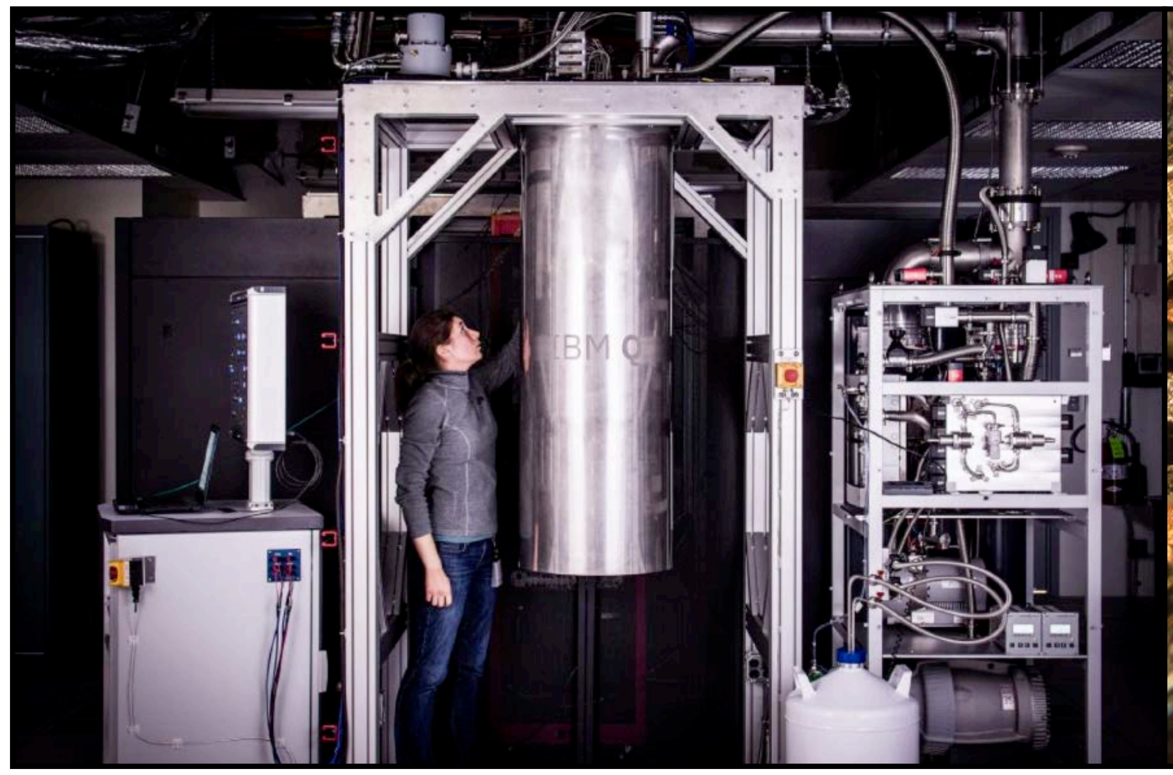


Delft

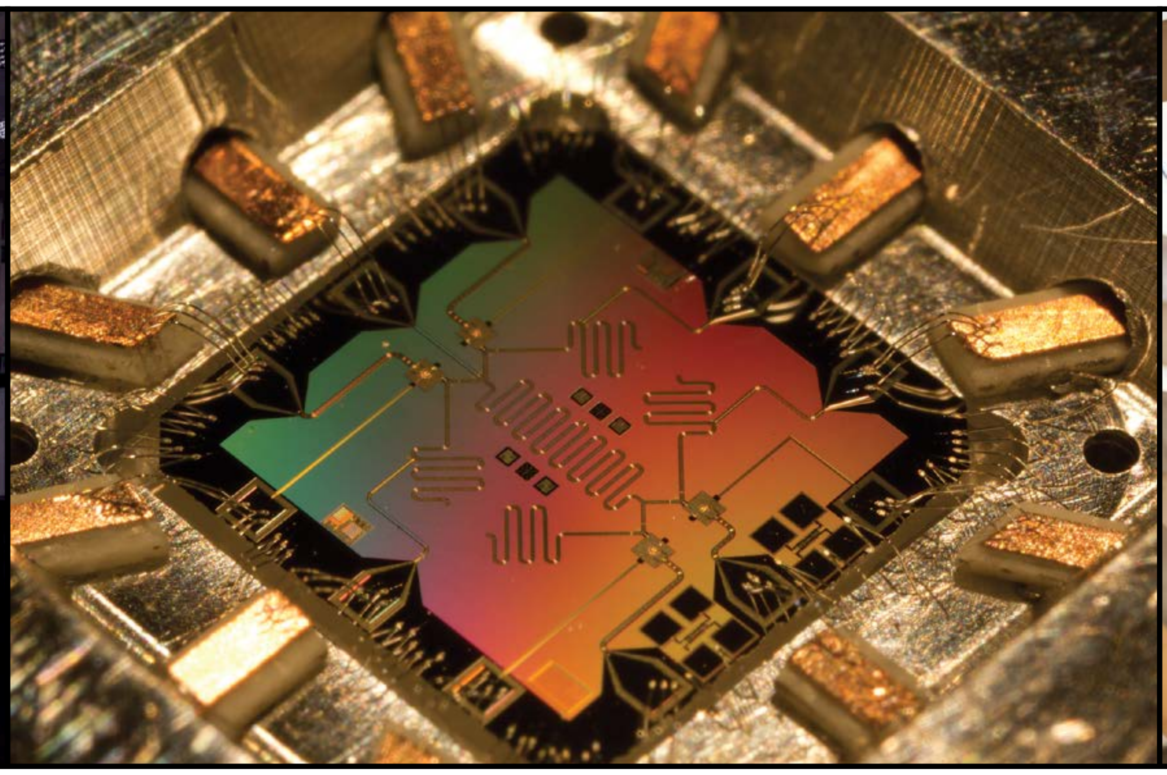


Maryland

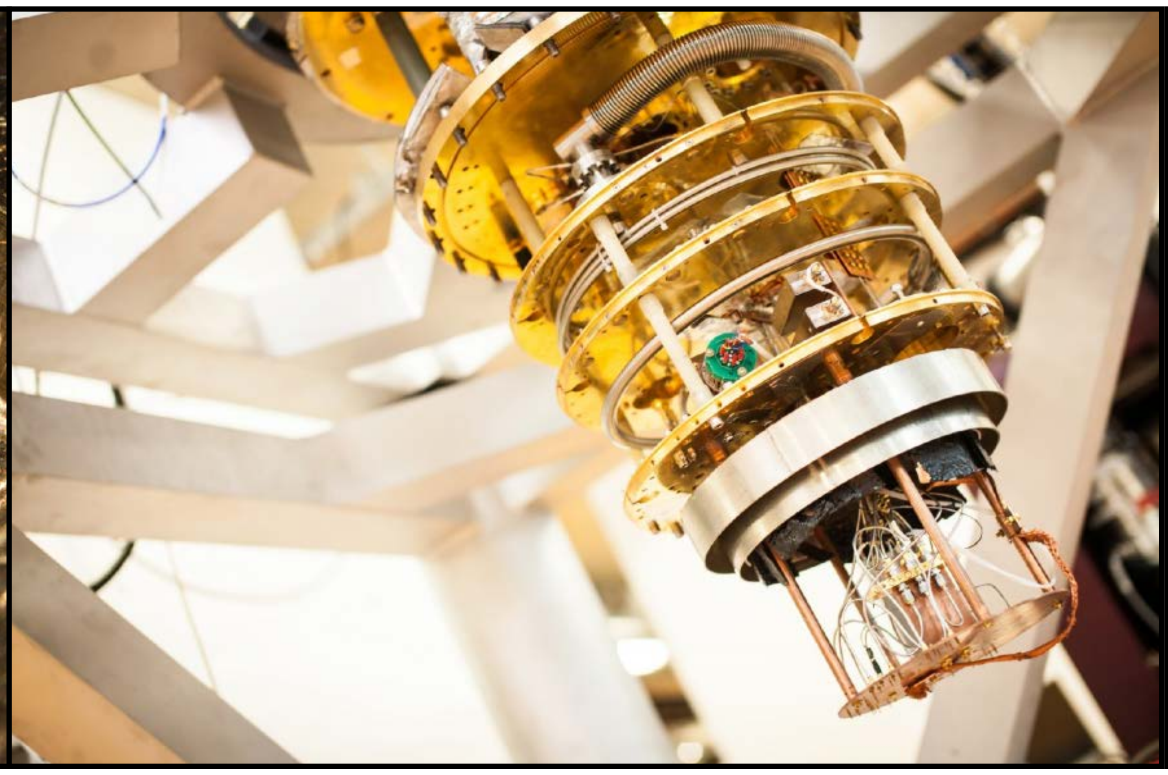
7. Making quantum simulation practical



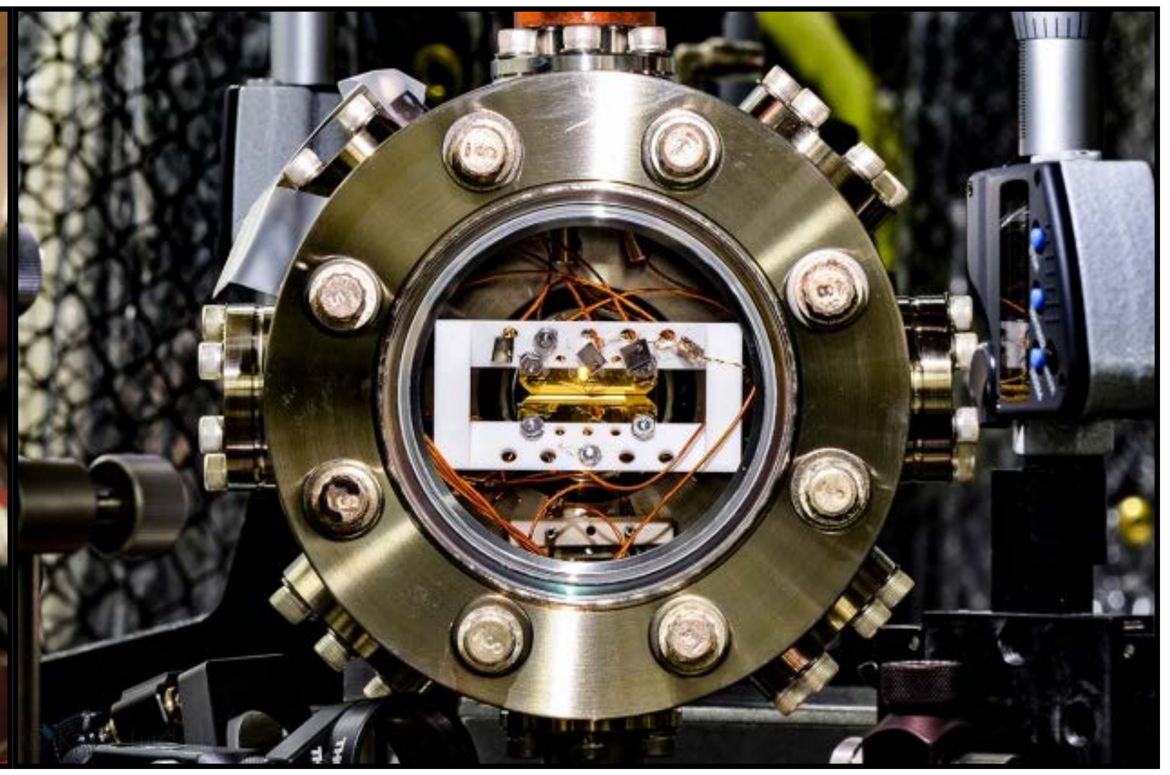
IBM



Google/UCSB



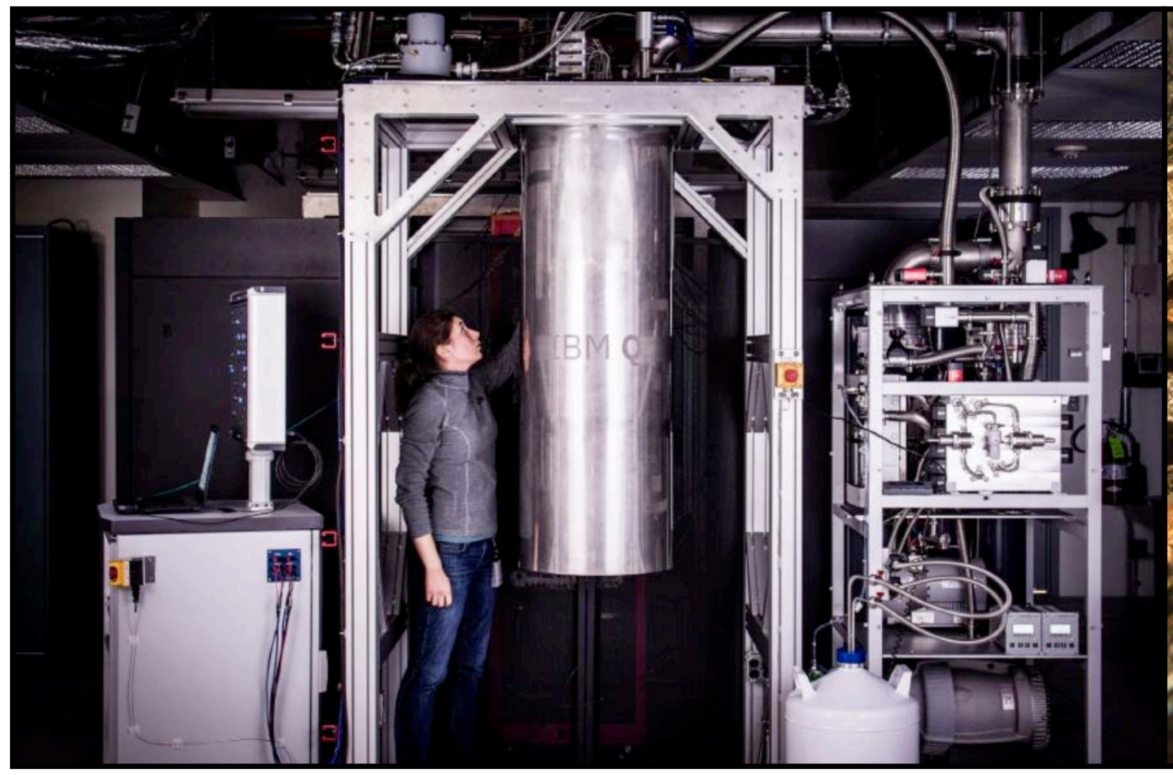
Delft

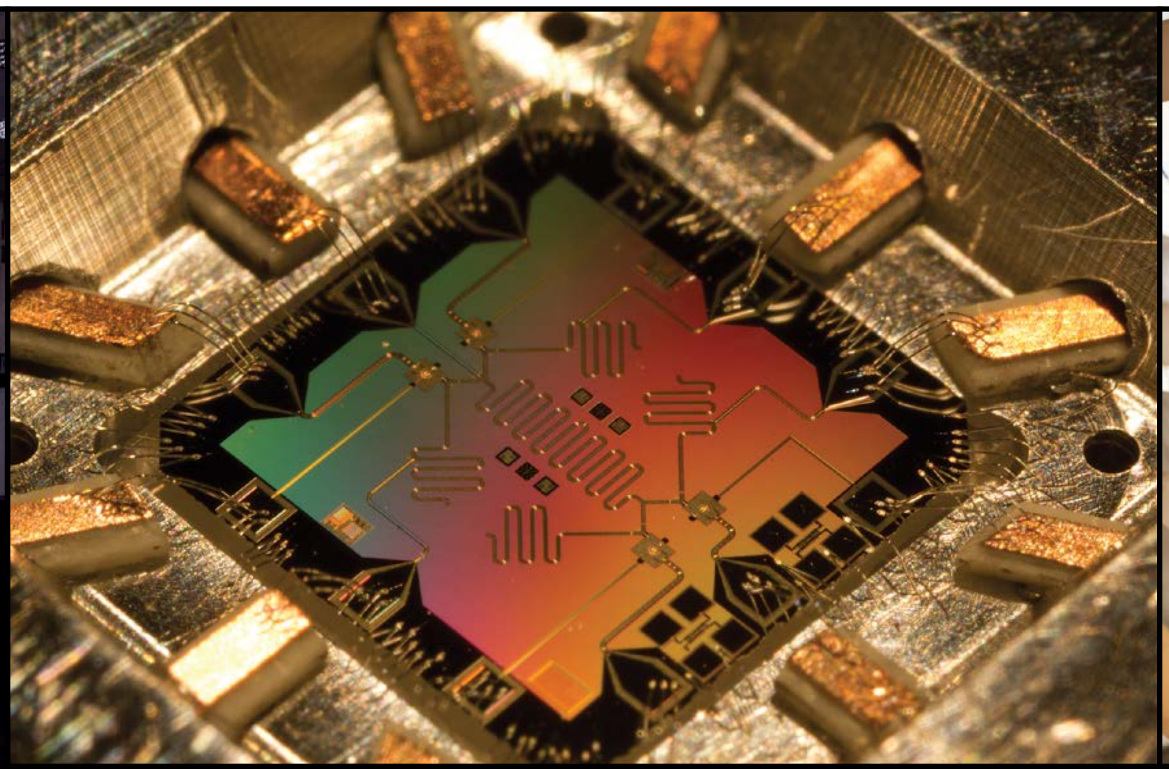


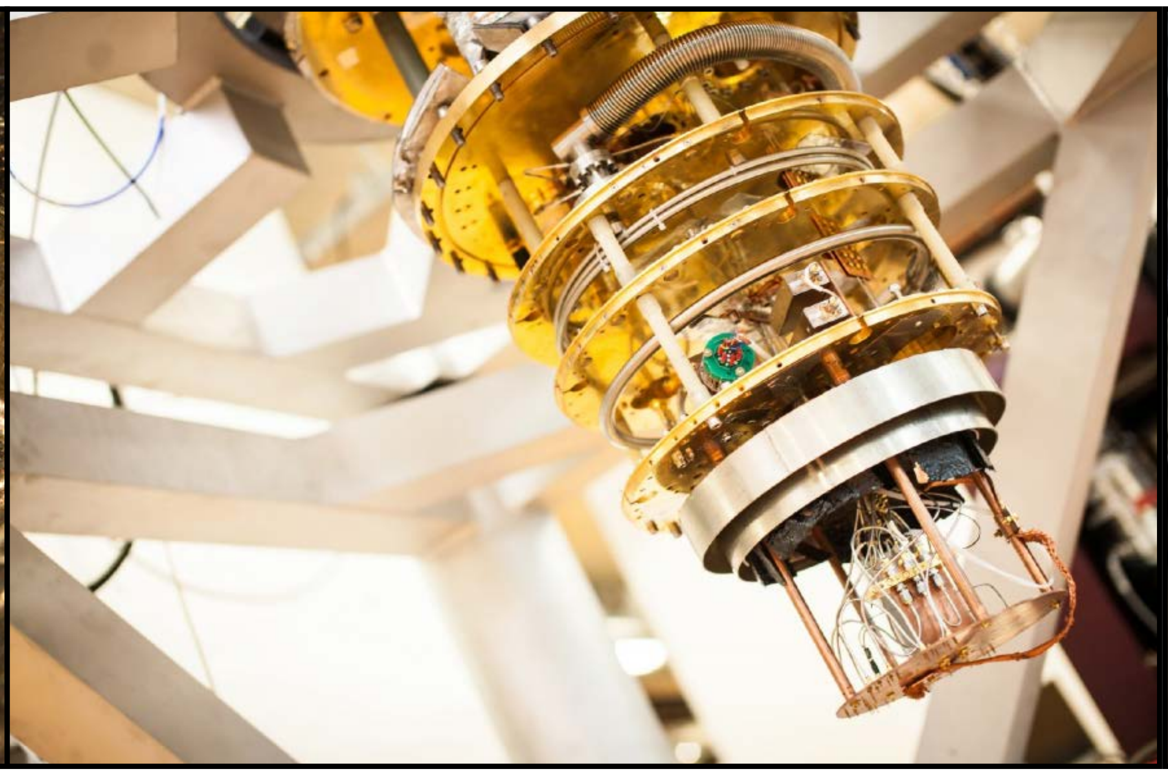
Maryland

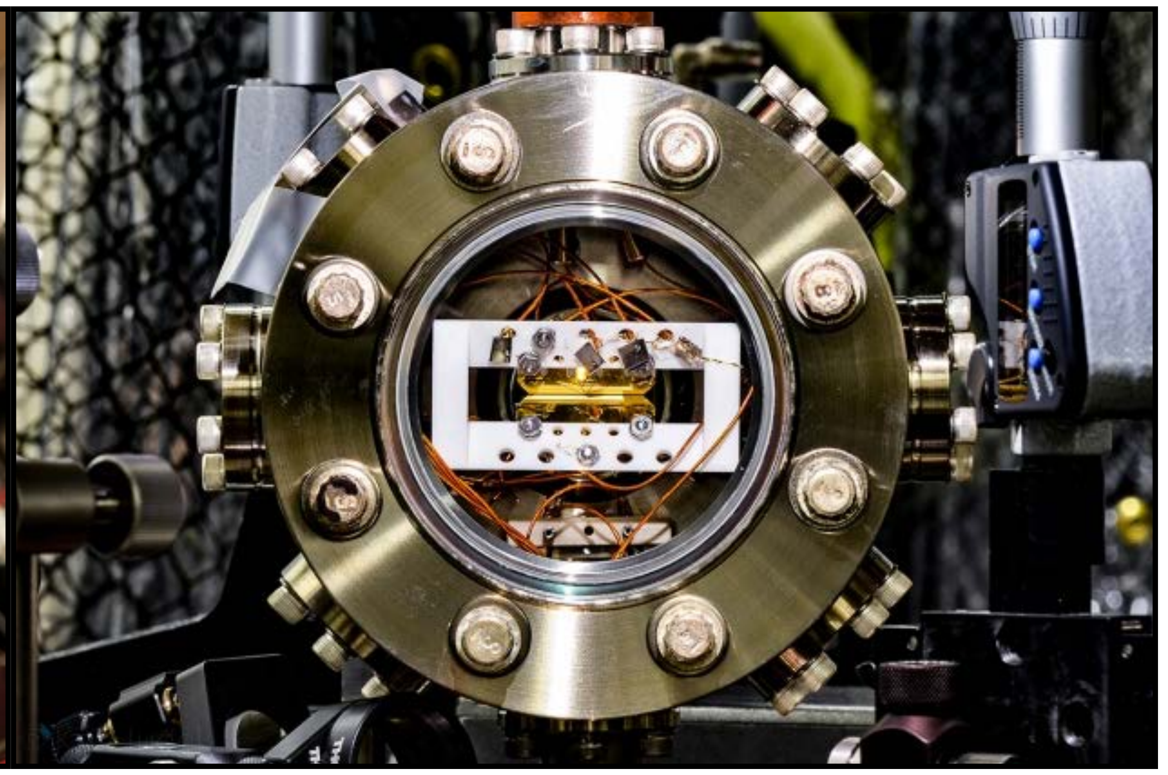
When can we use quantum computers to solve problems that are beyond the reach of classical computers?

7. Making quantum simulation practical


IBM


Google/UCSB


Delft

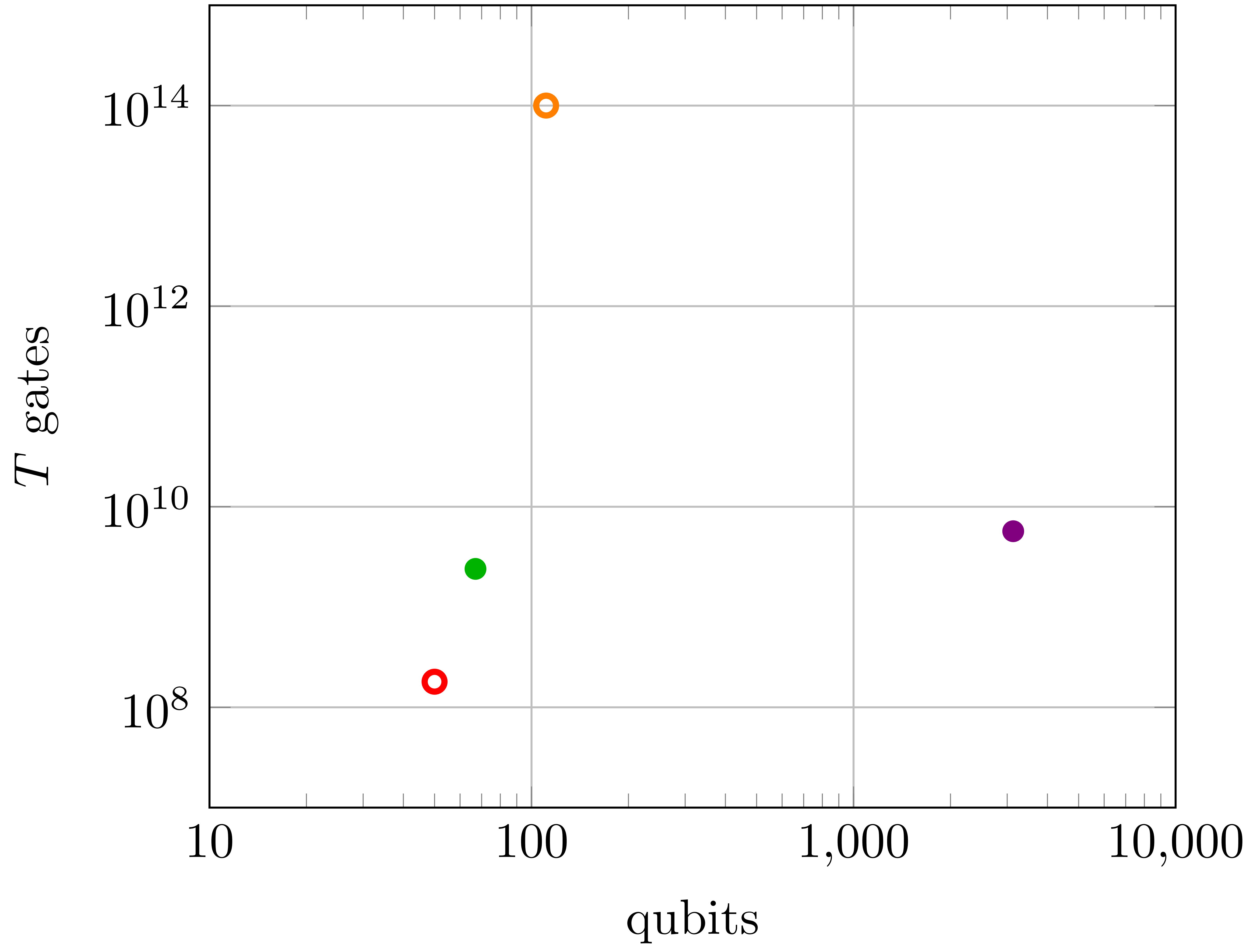

Maryland

When can we use quantum computers to solve problems that are beyond the reach of classical computers?

Challenges

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

7. Making quantum simulation practical



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 17]

Outlook

Outlook

Interplay between physics and computer science

- Ideas from CS: distributed graph coloring, query complexity lower bounds, Markov chains, …
- Ideas from physics: composite pulses, Lieb-Robinson bounds, scattering, …
- Quantum simulation algorithms will be powerful computational tools for answering questions about quantum physics

Outlook

Interplay between physics and computer science

- Ideas from CS: distributed graph coloring, query complexity lower bounds, Markov chains, …
- Ideas from physics: composite pulses, Lieb-Robinson bounds, scattering, …
- Quantum simulation algorithms will be powerful computational tools for answering questions about quantum physics

Ongoing challenges

- Find faster quantum simulation algorithms that exploit system structure
- Develop efficient practical implementations
- Use real quantum computers to do science!