MACQUARIE UNIVERSITY

Quantum algorithms for Hamiltonian simulation

Dominic Berry

Macquarie University

Richard Cleve

Andrew Childs

IQC

+

Rolando Somma

LANL

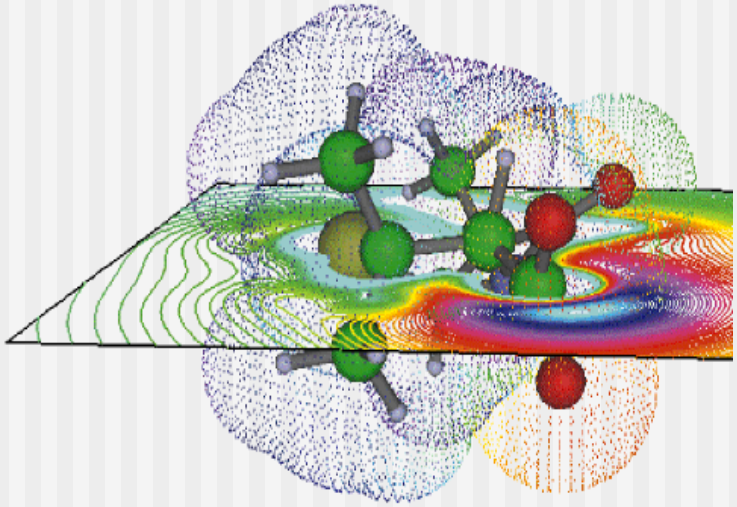
Sev Gharibian

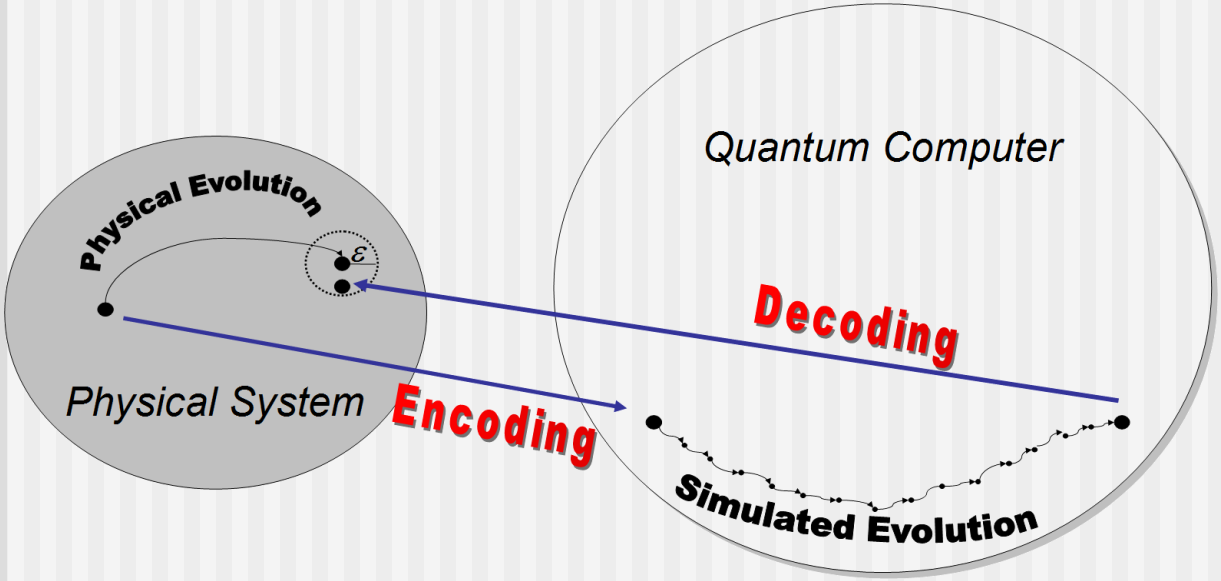
Berkeley

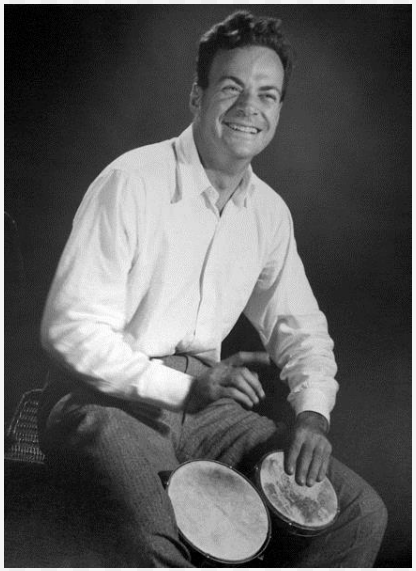
IQC Institute for Quantum Computing

Simulation of Hamiltonians

- Quantum computers could give an *exponential* speedup in the simulation of quantum physical systems.
- This is the original reason why Feynman proposed the idea of quantum computers.
- The state of the system is encoded into the quantum computer.







Richard Feynman

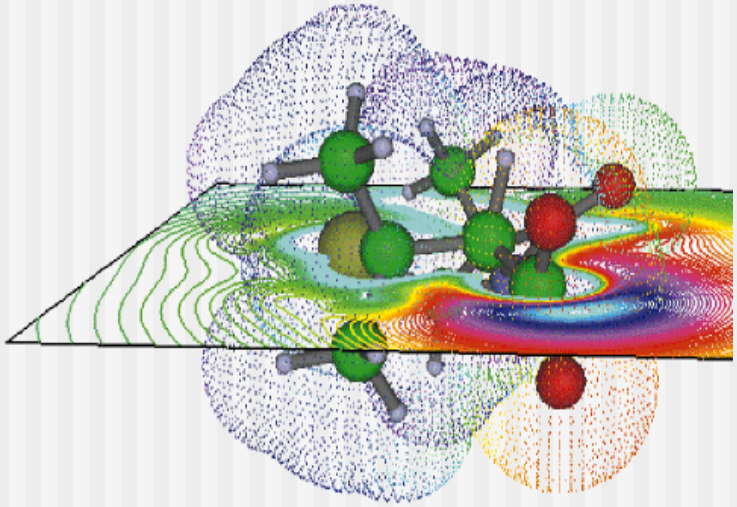
Simulation of Hamiltonians

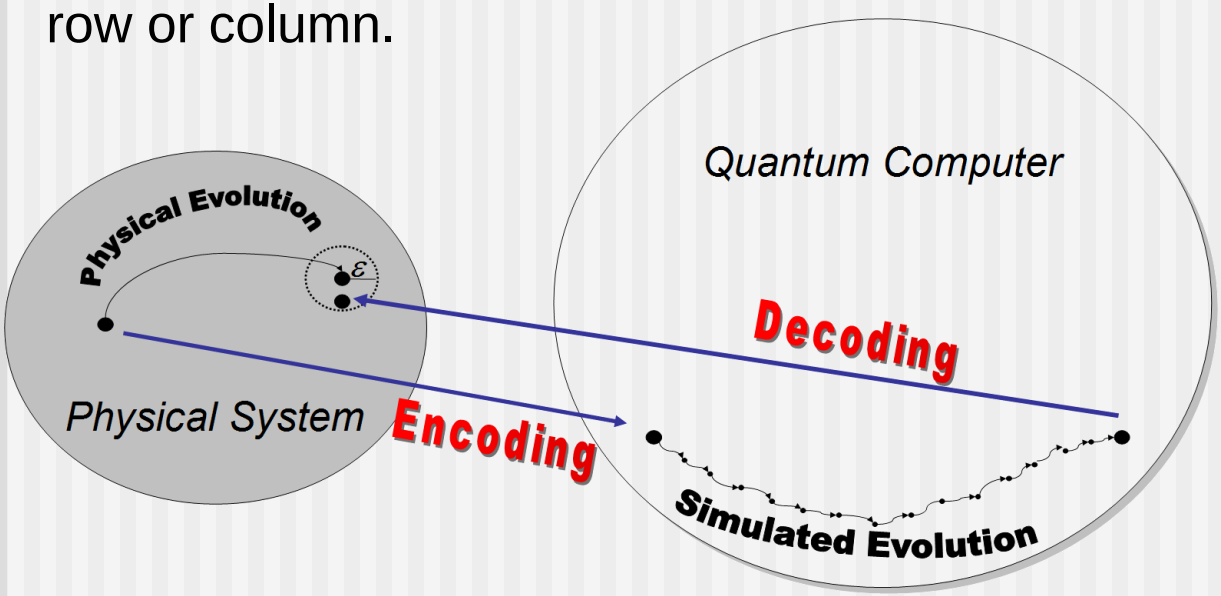
- Two scenarios:

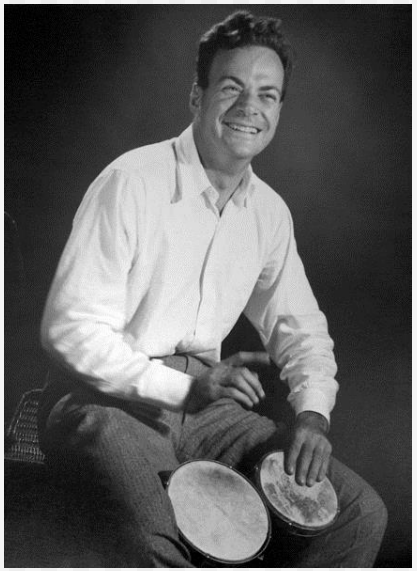
1. The Hamiltonian is given as a sum of interaction terms:

$$H = \sum_j H_j$$

2. The Hamiltonian is sparse, in that it has no more than d nonzero elements in any row or column.







Richard Feynman

Standard Methods

- Decompose the Hamiltonian as

$$H = \sum_{k=1}^{M} H_k$$

- The individual Hamiltonians H_k can be limited-dimension interaction Hamiltonians (Lloyd, 1996).
- Approximate evolution for short time as

$$e^{-iHT} = \prod_{k=1}^{M} e^{-iH_k T}$$

- For longer times, we divide the time up into many short times

$$e^{-iHT} = \left(\prod_{k=1}^{M} e^{-iH_k T/r} \right)^r$$

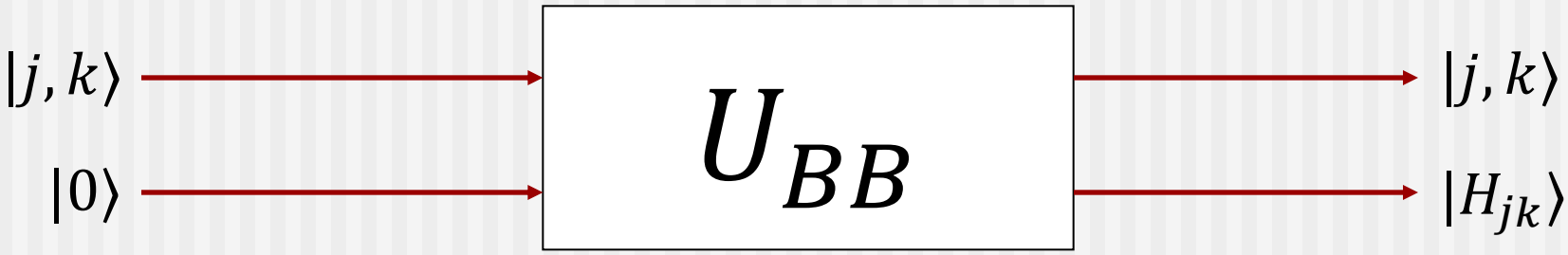
S. Lloyd, Science **273**, 1073 (1996).

Standard Methods

- More generally, we would like to be able to simulate *sparse* Hamiltonians.

$$H = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & \sqrt{2}i & \cdots & 0 \\ 0 & 3 & 0 & 0 & 0 & 1/2 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & \cdots & -\sqrt{3}+i \\ 0 & 0 & 0 & 1 & e^{i\pi/7} & 0 & \cdots & 0 \\ 0 & 0 & 0 & e^{-i\pi/7} & 2 & 0 & \cdots & 0 \\ -\sqrt{2}i & 1/2 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & -\sqrt{3}-i & 0 & 0 & 0 & \cdots & 1/10 \end{pmatrix}$$

- Positions and values of non-zero elements are given by oracle.



- This enables application to many other problems.

Standard Methods

- The individual H_k can be 2-sparse Hamiltonians obtained by a decomposition technique (Aharonov & Ta-Shma, 2003).
- Efficiency can be increased by improved decomposition techniques (Berry, Ahokas, Cleve, Sanders, 2007; Childs & Kothari, 2010).
- Higher-order decomposition formulae can also be used to obtain greater efficiency (Berry, Ahokas, Cleve, Sanders, 2007).

D. Aharonov and Ta-Shma, STOC 2003; quant-ph/0301023 (2003).
D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Comm. Math. Phys. **270**, 359 (2007).
A. M. Childs and R. Kothari, TQC 2010, Lecture Notes in Computer Science **6519**, 94 (2011).

Quantities involved in simulation

We want to simulation quantum evolution under a Hamiltonian

$$\frac{d}{dt}|\psi\rangle = -iH(t)|\psi\rangle$$

- ε – allowable error in the simulation
- T – time of evolution under the Hamiltonian
- d – sparseness, i.e. maximum number of nonzero elements
- $\|H\|$ – norm of the Hamiltonian to be simulated
- $\|H'\|$ – norm of the time-derivative of the Hamiltonian
- N – dimension of the system

Standard Methods - Limitations

- The scaling is always polynomial in the allowable error, ε.
- The scaling for time-dependent Hamiltonians depends heavily on the derivatives of the Hamiltonian.

- The scaling in T is always superlinear, whereas lower bound is linear in T.
- The scaling is at best d^3 in the sparseness.

Part 1: Polylog Hamiltonian simulation

How can we perform simulation polylog in the error?

Known results

- It is possible to decompose a sparse Hamiltonian into $O(d^2)$ 1-sparse Hamiltonians with complexity $O(\log^* n)$ [1].
- This can be improved to $O(d)$ Hamiltonians, at the cost of complexity linear in d [2].
- Arbitrary order Lie-Trotter-Suzuki formulae can be used to obtain scaling as $O((\|H\|T)^{1+1/2k})$ for arbitrarily large integer k. The scaling in terms of the allowable error is $O(1/\varepsilon^{1/2k})$ [1].
- Using quantum walks without a decomposition enables complexity strictly linear in $\|H\|T$, but as scaling in the error of $O(1/\sqrt{\varepsilon})$ [3].
- Similar scaling can be obtained for *time-dependent* Hamiltonians, but the complexity now depends on the higher-order derivatives [4].
- An algorithm with randomised times enables complexity independent of the derivatives of the Hamiltonian, at the expense of worse scaling in ε [5].

[1] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Comm. Math. Phys. **270**, 359 (2007).
[2] A. M. Childs and R. Kothari, TQC 2010, Lecture Notes in Computer Science **6519**, 94 (2011).
[3] D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).
[4] N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders, J. Phys. A: Math. Theor. **43**, 065203 (2010).
[5] D. Poulin, A. Qarry, R. D. Somma, and F. Verstraete, Phys. Rev. Lett. **106**, 170501 (2011).

Main result

- We obtain scaling as

$$O(d^2 \|H\| T \times \text{polylog})$$

- Advances over prior work:

1. The scaling is polylog in the error – something which has never been achieved before.
2. The scaling is only polylog in the derivative of the Hamiltonian – independence had been achieved before, but at the expense of poor scaling in ε and $\|H\|T$ [5].
3. The scaling is linear in $\|H\|T$ up to a polylog factor – linear scaling had been achieved before, but with poor scaling in ε [3].
4. The scaling in d is better than has been achieved before, except in [3].

[1] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Comm. Math. Phys. **270**, 359 (2007).
[2] A. M. Childs and R. Kothari, TQC 2010, Lecture Notes in Computer Science **6519**, 94 (2011).
[3] D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).
[4] N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders, J. Phys. A: Math. Theor. **43**, 065203 (2010).
[5] D. Poulin, A. Qarry, R. D. Somma, and F. Verstraete, Phys. Rev. Lett. **106**, 170501 (2011).

Main result

- We obtain scaling as

$$O(d^2 \|H\| T \times \mathrm{polylog})$$

- Method:

1. Decompose Hamiltonian into 1-sparse Hamiltonians as in [1].
2. Further decompose Hamiltonian into a sum of self-inverse Hamiltonians.
3. Use a low-order Lie-Trotter expansion of the evolution into a product of evolutions under these self-inverse Hamiltonians.
4. Use method from [2] to turn Trotter decomposition into discrete steps at a superposition of times.
5. Use method from [3] to perform measurements efficiently.

[1] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Comm. Math. Phys. **270**, 359 (2007).
[2] R. Cleve, D. Gottesman, M. Mosca, R. Somma, and D. Yonge-Mallo, In Proc. 41st ACM Symposium on Theory of Computing, pp. 409-416 (2009).
[3] D. W. Berry, R. Cleve, and S. Gharibian, arXiv:1211.4637 (2012).

1. Decompose Hamiltonian to 1-sparse

- Sparse Hamiltonian has no more than d nonzero elements in any row or column, e.g. $d = 2$

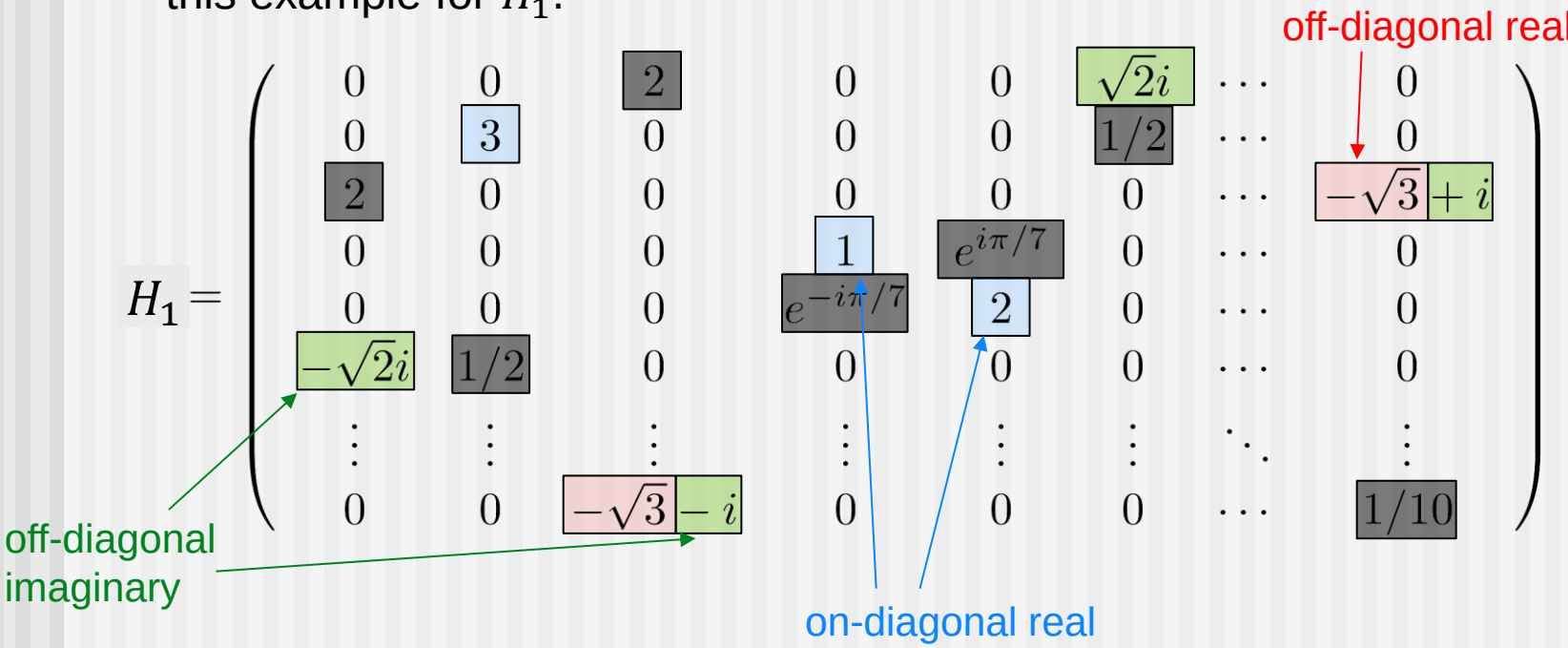
$$H = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & \sqrt{2}i & \cdots & 0 \\ 0 & 3 & 0 & 0 & 0 & 1/2 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & \cdots & -\sqrt{3}+i \\ 0 & 0 & 0 & 1 & e^{i\pi/7} & 0 & \cdots & 0 \\ 0 & 0 & 0 & e^{-i\pi/7} & 2 & 0 & \cdots & 0 \\ -\sqrt{2}i & 1/2 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & -\sqrt{3}-i & 0 & 0 & 0 & \cdots & 1/10 \end{pmatrix}$$

- A 1-sparse Hamiltonian has no more than one nonzero element.
- We could decompose Hamiltonian into H_1 and H_2 shown in blue and yellow.
- In general we can decompose into $6d^2$ parts.

D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Comm. Math. Phys. **270**, 359 (2007).

2. Decompose 1-sparse to self-inverse

- We further divide the 1-sparse Hamiltonian into X, Y and Z components, in this example for H_1.



- The X and Y components are proportional to Pauli X and Y matrices in each 2×2 subspace.
- The Z component is a phase shift in a 1×1 subspace.

2. Decompose 1-sparse to self-inverse

- Consider just the X component. We further decompose it into components of magnitude $2\varepsilon_H$.

$$H_{1,X} = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & \sqrt{2}i & \cdots & 0 \\ 0 & 3 & 0 & 0 & 0 & 1/2 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & \cdots & -\sqrt{3}+i \\ 0 & 0 & 0 & 1 & e^{i\pi/7} & 0 & \cdots & 0 \\ 0 & 0 & 0 & e^{-i\pi/7} & 2 & 0 & \cdots & 0 \\ -\sqrt{2}i & 1/2 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & -\sqrt{3}-i & 0 & 0 & 0 & \cdots & 1/10 \end{pmatrix}$$

- Take $\varepsilon_H = 1/4$. Then we can approximate

$$-\sqrt{3} \approx -\frac{1}{2}\boxed{-\frac{1}{2}}-\frac{1}{2}+0$$

take component 2

2. Decompose 1-sparse to self-inverse

- Consider just the X component. We further decompose it into components of magnitude $2\varepsilon_H$.

$$H_{1,X,2} = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & \sqrt{2}i & \cdots & 0 \\ 0 & 3 & 0 & 0 & 0 & 1/2 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & \cdots & -1/2 + i \\ 0 & 0 & 0 & 1 & e^{i\pi/7} & 0 & \cdots & 0 \\ 0 & 0 & 0 & e^{-i\pi/7} & 2 & 0 & \cdots & 0 \\ -\sqrt{2}i & 1/2 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & -1/2 - i & 0 & 0 & 0 & \cdots & 1/10 \end{pmatrix}$$

- Take $\varepsilon_H = 1/4$. Then we can approximate

$$-\sqrt{3} \approx -\frac{1}{2} \boxed{-\frac{1}{2}} -\frac{1}{2} + 0$$

take component 2

2. Decompose 1-sparse to self-inverse

- To obtain self-inverse matrices, we want $+1$ or -1 to appear in each column once.

$$H_{1,X,2} = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & \sqrt{2}i & \cdots & 0 \\ 0 & 3 & 0 & 0 & 0 & 1/2 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & \cdots & -1/2 + i \\ 0 & 0 & 0 & 1 & e^{i\pi/7} & 0 & \cdots & 0 \\ 0 & 0 & 0 & e^{-i\pi/7} & 2 & 0 & \cdots & 0 \\ -\sqrt{2}i & 1/2 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & -1/2 - i & 0 & 0 & 0 & \cdots & 1/10 \end{pmatrix}$$

- We further expand

$$\frac{1}{2} = \frac{1}{4} + \frac{1}{4} \qquad -\frac{1}{2} = -\frac{1}{4} - \frac{1}{4} \qquad 0 = \frac{1}{4} - \frac{1}{4}$$

2. Decompose 1-sparse to self-inverse

- To obtain self-inverse matrices, we want $+1$ or -1 to appear in each column once.

$$H_{1,X,2,+} = \begin{pmatrix} 1/4 & 0 & 2 & 0 & 0 & \sqrt{2}i & \cdots & 0 \\ 0 & 1/4 & 0 & 0 & 0 & 1/2 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & \cdots & -1/4 + i \\ 0 & 0 & 0 & 1/4 & e^{i\pi/7} & 0 & \cdots & 0 \\ 0 & 0 & 0 & e^{-i\pi/7} & 1/4 & 0 & \cdots & 0 \\ -\sqrt{2}i & 1/2 & 0 & 0 & 0 & 1/4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & -1/4 - i & 0 & 0 & 0 & \cdots & 1/10 \end{pmatrix}$$

- We further expand

take first component

$$\frac{1}{2} = \frac{1}{4} + \frac{1}{4} \qquad -\frac{1}{2} = -\frac{1}{4} - \frac{1}{4} \qquad 0 = \frac{1}{4} - \frac{1}{4}$$

- To make it 1-sparse we fill in on the diagonal as needed.

2. Decompose 1-sparse to self-inverse

- To obtain self-inverse matrices, we want $+1$ or -1 to appear in each column once.

$$H_{1,X,2,+} = \begin{pmatrix} 1 & 0 & 2 & 0 & 0 & \sqrt{2}i & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & 1/2 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & \cdots & -1+i \\ 0 & 0 & 0 & 1 & e^{i\pi/7} & 0 & \cdots & 0 \\ 0 & 0 & 0 & e^{-i\pi/7} & 1 & 0 & \cdots & 0 \\ -\sqrt{2}i & 1/2 & 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & -1-i & 0 & 0 & 0 & \cdots & 1/10 \end{pmatrix} \varepsilon_H$$

- We further expand

take first component

$$\frac{1}{2} = \frac{1}{4} + \frac{1}{4} \qquad -\frac{1}{2} = -\frac{1}{4} - \frac{1}{4} \qquad 0 = \frac{1}{4} - \frac{1}{4}$$

- To make it 1-sparse we fill in on the diagonal as needed.

3. Trotter expansion

- The Hamiltonian evolution is

$$\exp(-i(H_1 + H_2)T)$$

- More generally time-dependent evolution

time ordering → $\mathscr{T} \exp\left[-i \int_0^T (H_1(t) + H_2(t))dt\right]$

- This can be thought of as the limit of a large number, r, of small intervals.

$$\lim_{r\to\infty} \prod_{j=1}^{r} e^{-iH_1(t_j)\delta t} e^{-iH_2(t_j)\delta t} \qquad t_j = j\delta t, \quad \delta t = T/r$$

- We can approximate the time evolution using finite r. The error scales as

$$O\left(\frac{(\Lambda T)^2}{r}\right) \qquad \Lambda = \max(\|H\|, \|H'\|)$$

- To bound error by ε, can use

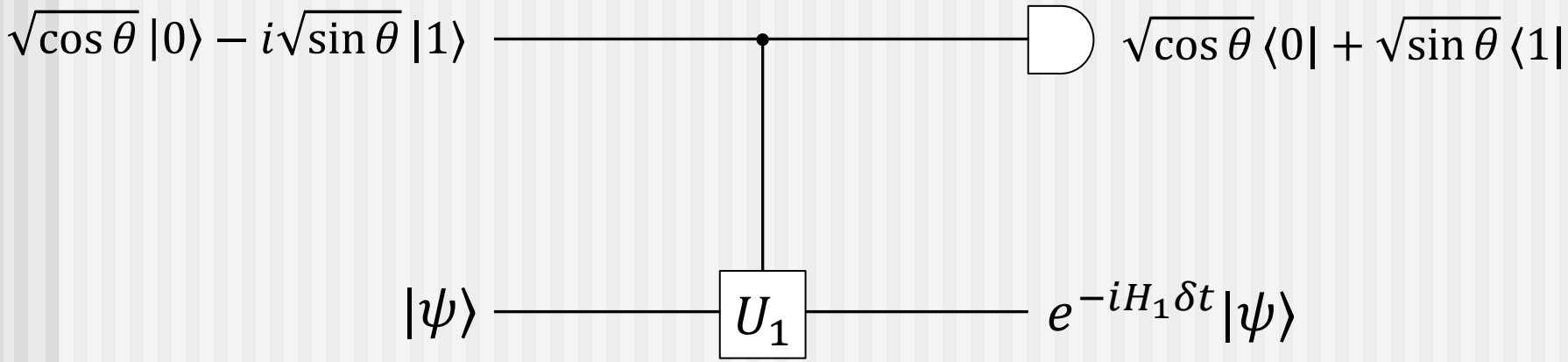
$$r \propto \frac{(\Lambda T)^2}{\varepsilon}$$

4. Using CGMSY'09 technique

- $H_1 = \varepsilon_H U_1$, where U_1 is self-inverse, so

$$e^{-iH_1\delta t} = \cos\theta - iU_1 \sin\theta \qquad \theta = \varepsilon_H \delta t$$

- Implement the operation probabilistically with a control qubit.



R. Cleve, D. Gottesman, M. Mosca, R. Somma, and D. Yonge-Mallo, In Proc. 41st ACM Symposium on Theory of Computing, pp. 409-416 (2009).

4. Using CGMSY'09 technique

- $H_1 = \varepsilon_H U_1$, where U_1 is self-inverse, so

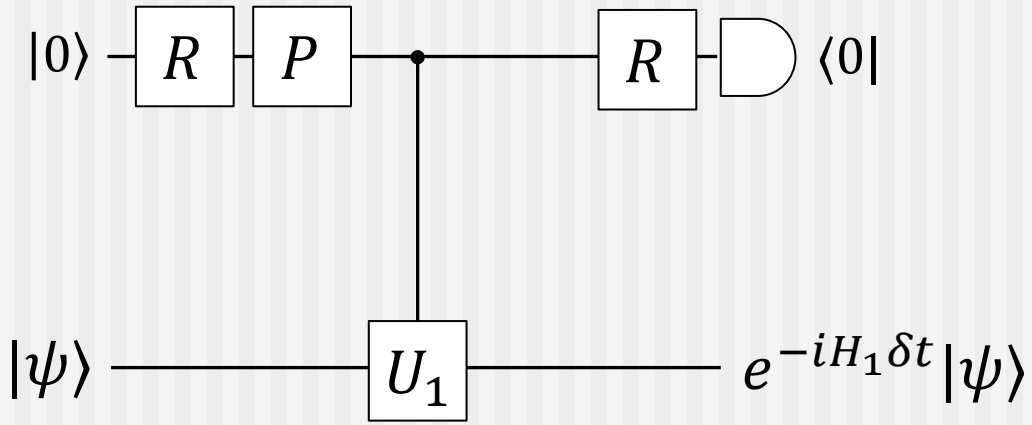
$$e^{-iH_1\delta t} = \cos\theta - iU_1 \sin\theta \qquad \theta = \varepsilon_H \delta t$$

- Implement the operation probabilistically with a control qubit.

$$R = \begin{pmatrix} \alpha & \beta \\ \beta & -\alpha \end{pmatrix}$$

$$\beta = \sqrt{\sin\theta}$$

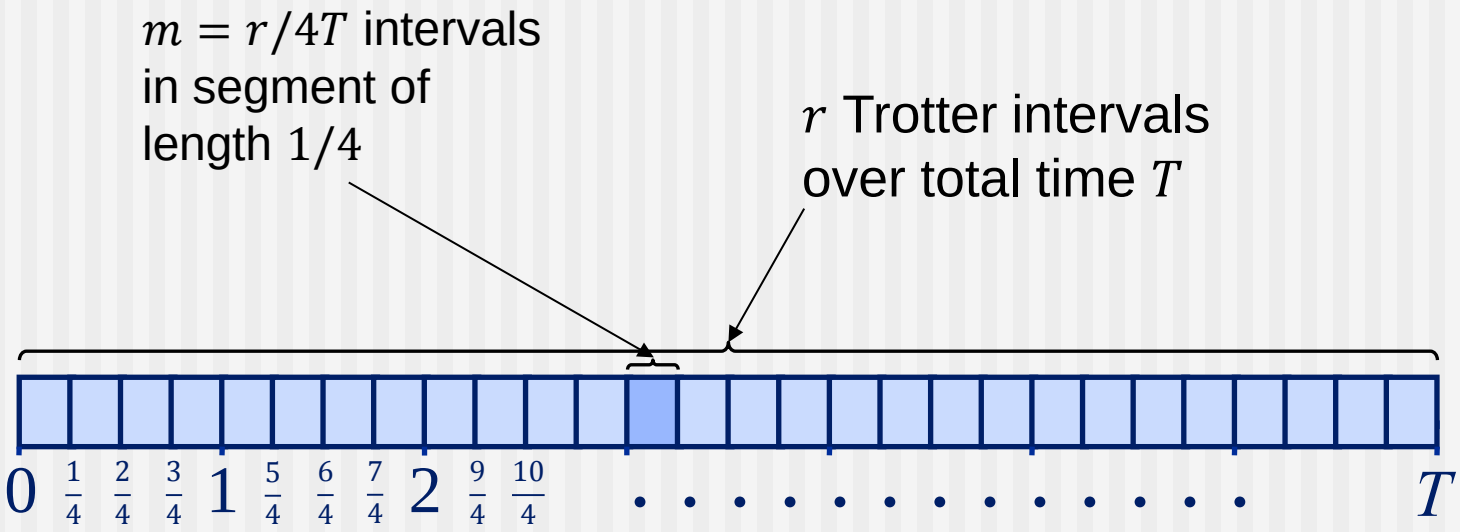
$$P = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$$



R. Cleve, D. Gottesman, M. Mosca, R. Somma, and D. Yonge-Mallo, In Proc. 41st ACM Symposium on Theory of Computing, pp. 409-416 (2009).

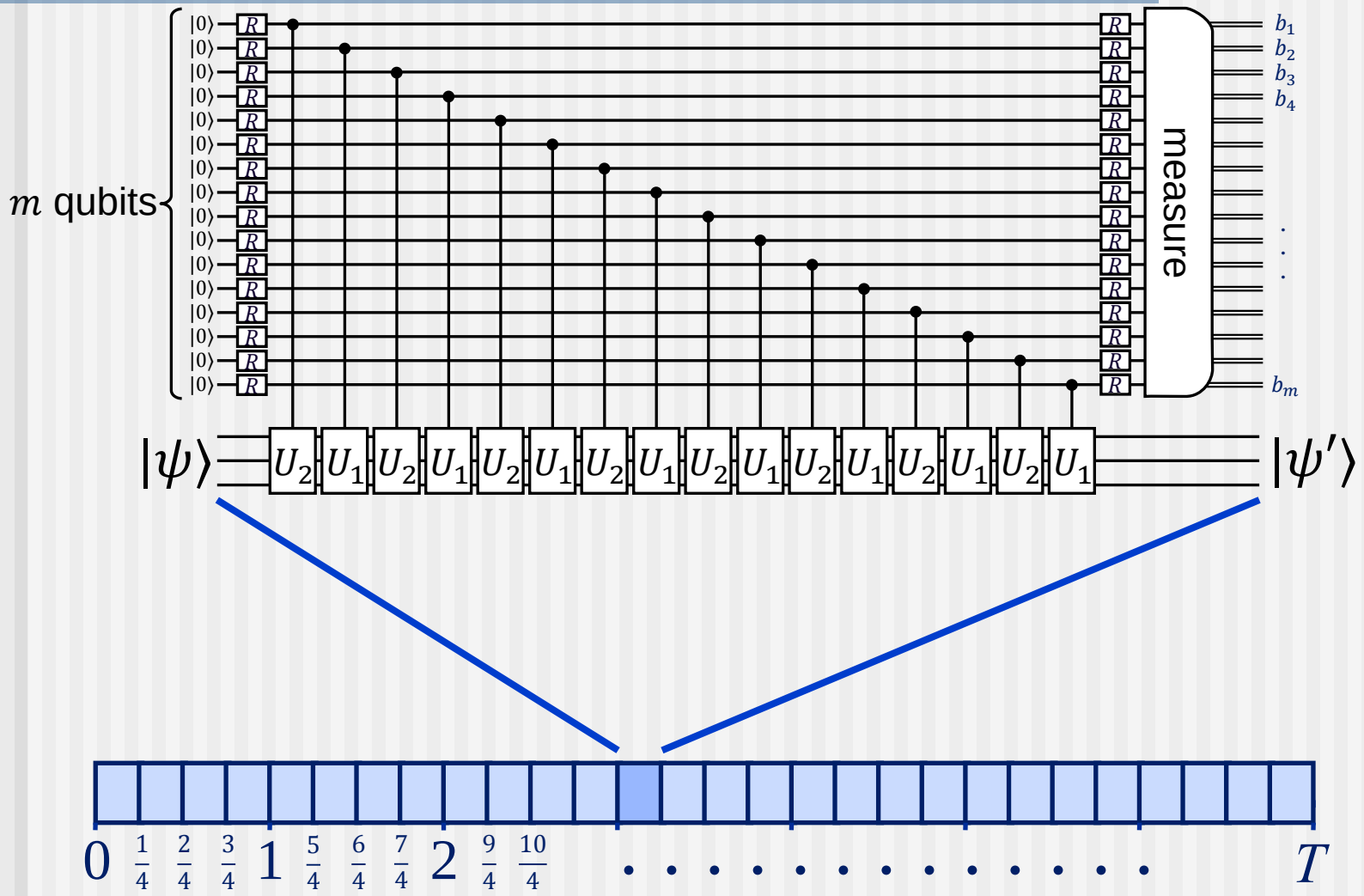
4. Using CGMSY'09 technique

Dividing the time into segments of length $1/4$, the probability of success for any segment is $\geq 3/4$.



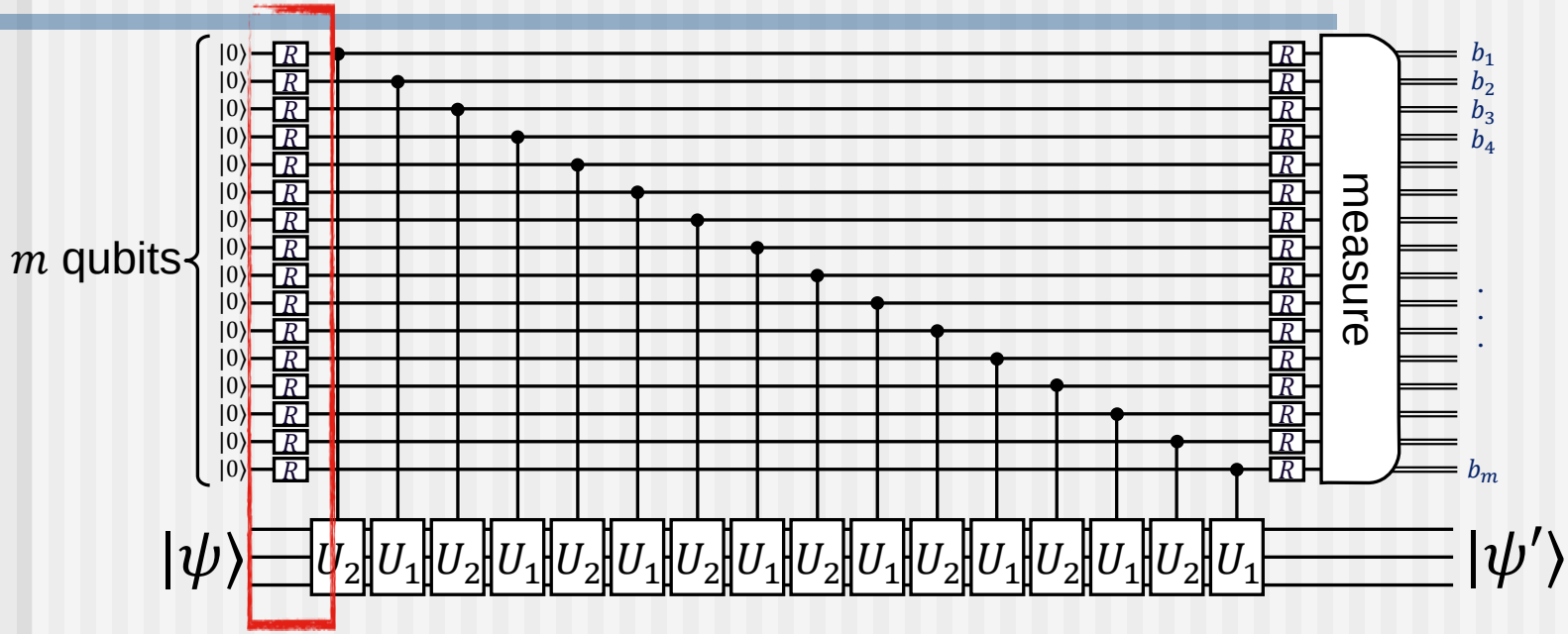
R. Cleve, D. Gottesman, M. Mosca, R. Somma, and D. Yonge-Mallo, In Proc. 41st ACM Symposium on Theory of Computing, pp. 409-416 (2009).

4. Using CGMSY'09 technique



R. Cleve, D. Gottesman, M. Mosca, R. Somma, and D. Yonge-Mallo, In Proc. 41st ACM Symposium on Theory of Computing, pp. 409-416 (2009).

4. Using CGMSY'09 technique



$$R^{\otimes m} = \overbrace{R \otimes R \otimes \cdots \otimes R}^{m} \text{ where } R = \begin{pmatrix} \alpha & \beta \\ \beta & -\alpha \end{pmatrix}$$

$$\alpha \approx 1$$

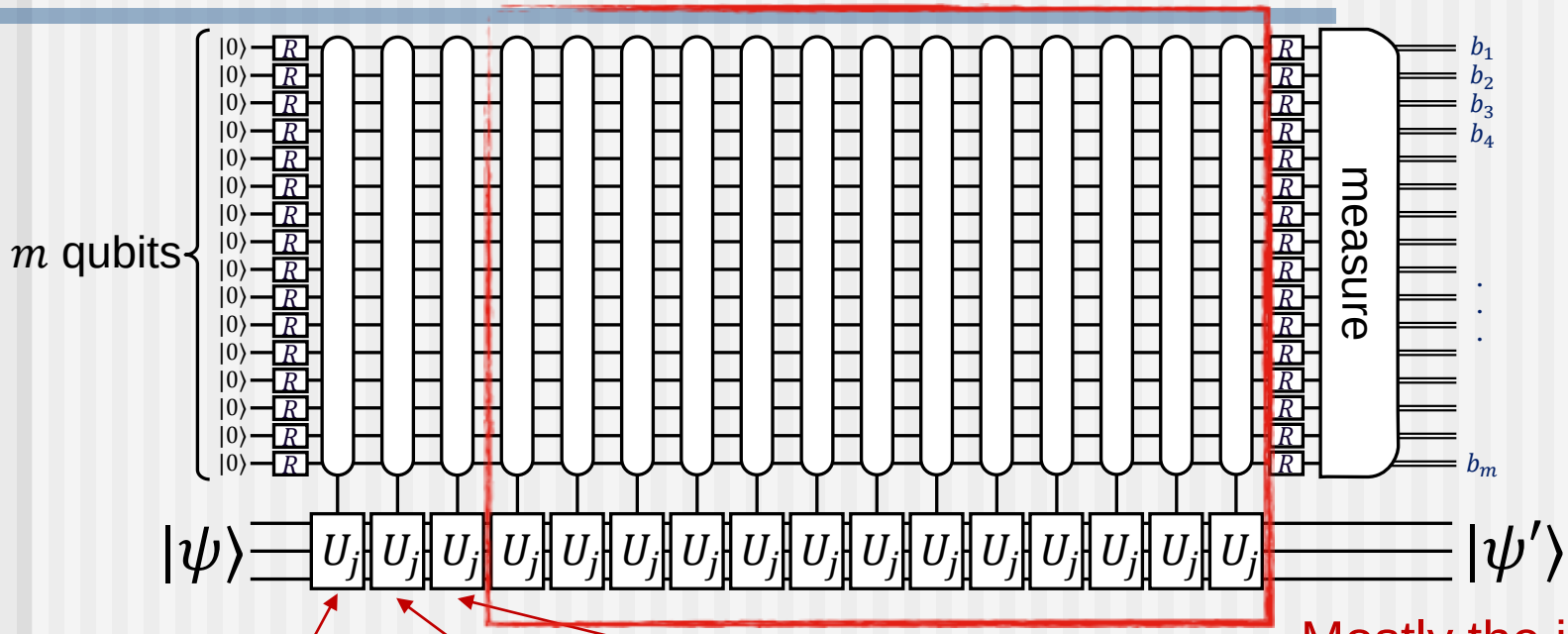
$$\beta \propto \frac{1}{\sqrt{m}}$$

$$R^{\otimes m}|0^m\rangle = (\alpha|0\rangle + \beta|1\rangle)^{\otimes m}$$

$$= \sum_x \alpha^{m-|x|}\beta^{|x|}|x\rangle$$

like Bernoulli trials, but in superposition

4. Using CGMSY'09 technique



Mostly the identity, so can be omitted.

$|0000000100000000010000000000001000000\rangle$

p_1

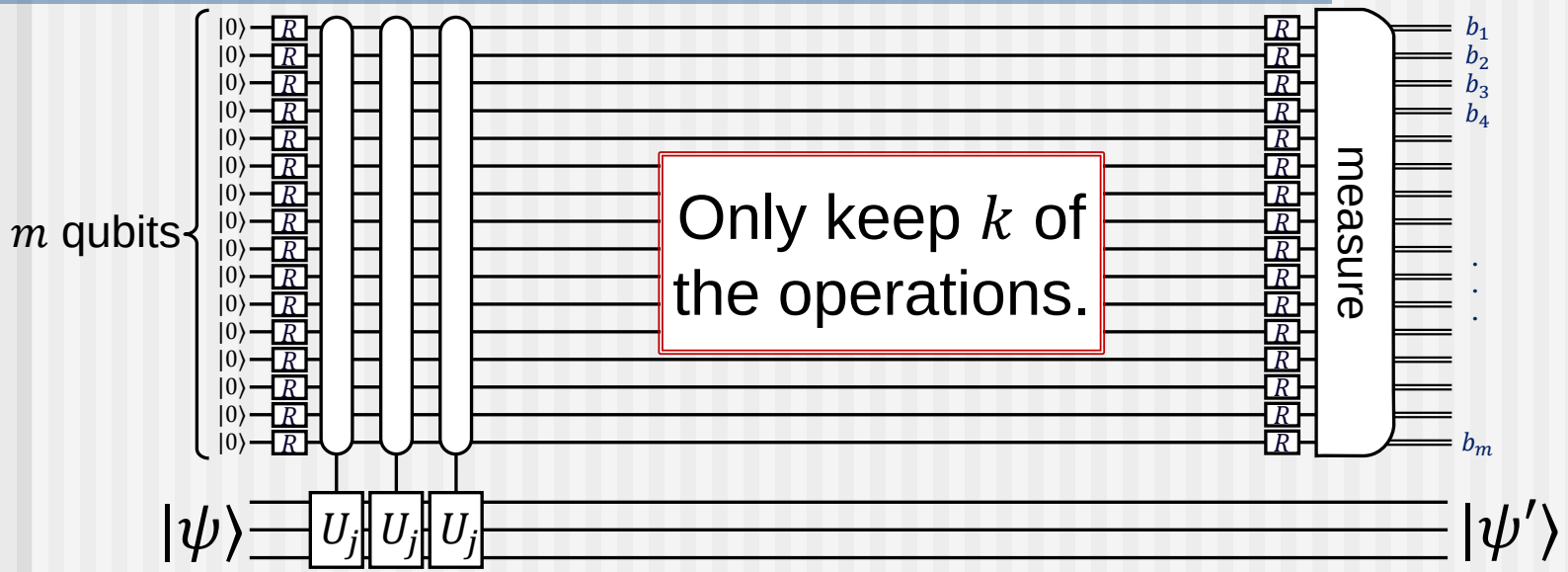
p_2

p_3

Each U_j is U_1 or U_2 depending on whether p_ℓ is even or odd.

If there are no more ones then U_j is the identity.

4. Using CGMSY'09 technique



$|0000000100000000010000000000001000000\rangle$

p_1

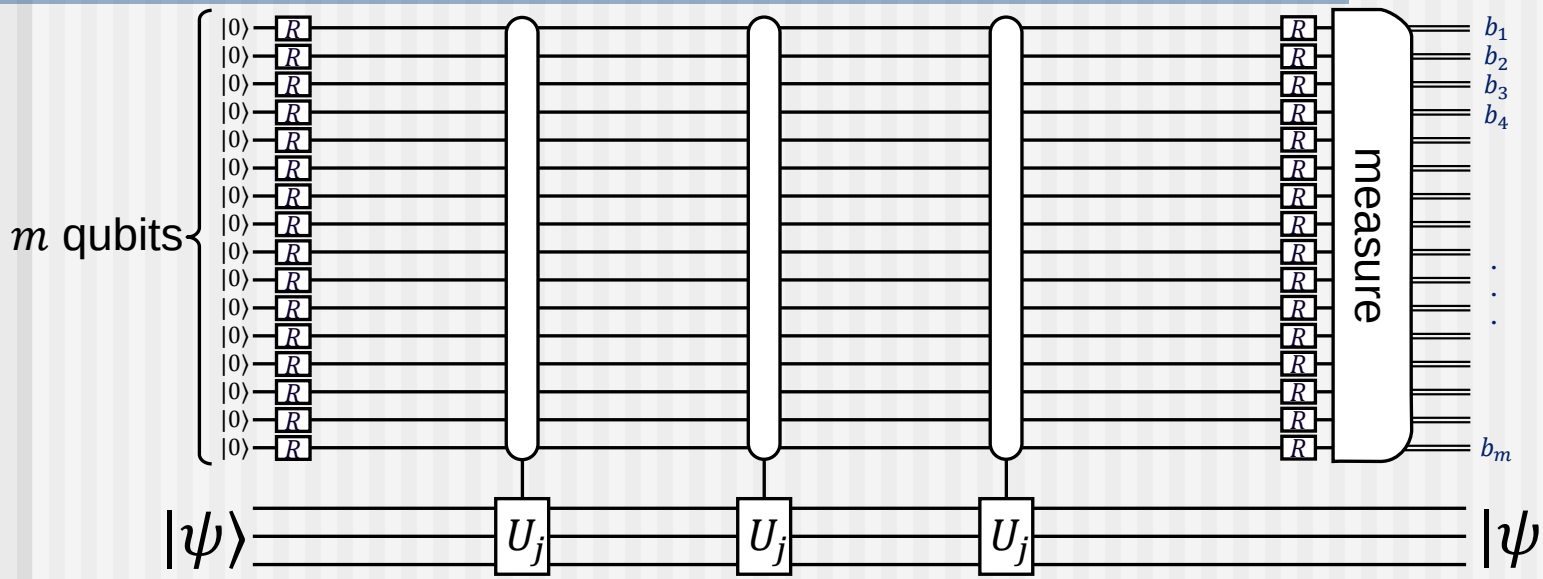
p_2

p_3

Each U_j is U_1 or U_2 depending on whether p_ℓ is even or odd.

If there are no more ones then U_j is the identity.

4. Using CGMSY'09 technique



$|0000000100000000010000000000001000000\rangle$

p_1

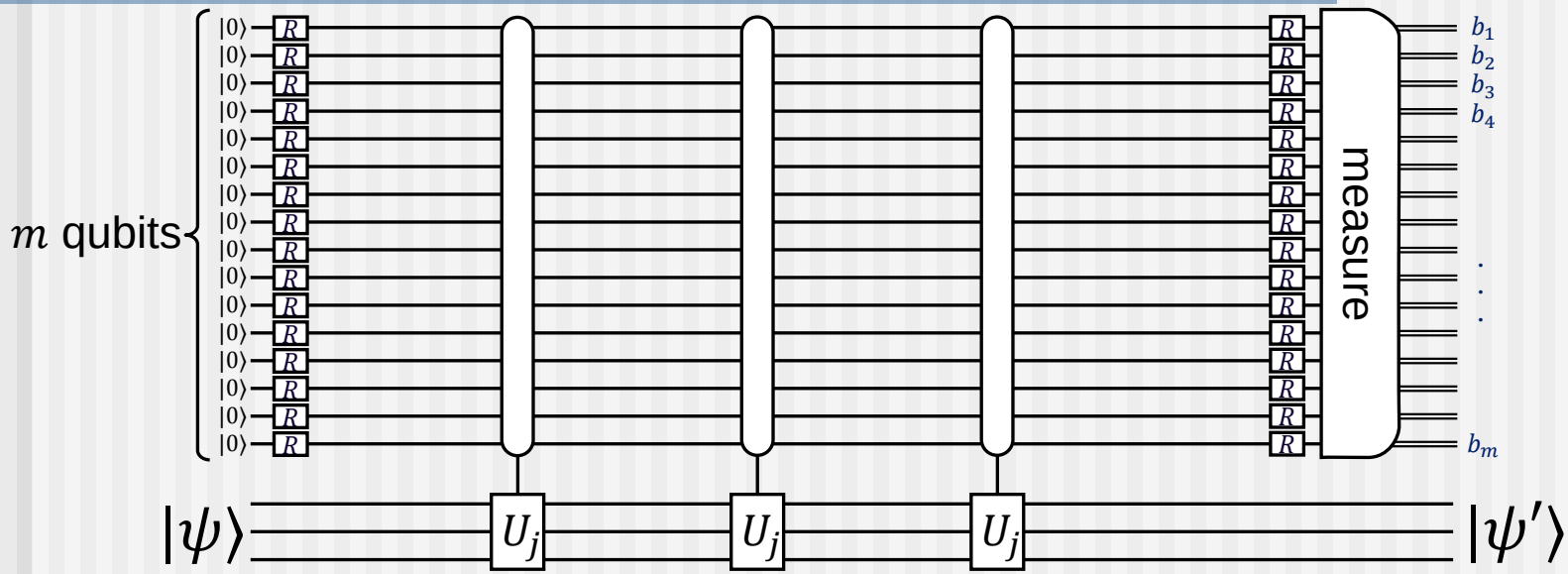
p_2

p_3

Each U_j is U_1 or U_2 depending on whether p_ℓ is even or odd.

If there are no more ones then U_j is the identity.

4. Using CGMSY'09 technique



$|0000000100000000010000000000001000000\rangle$

p_1

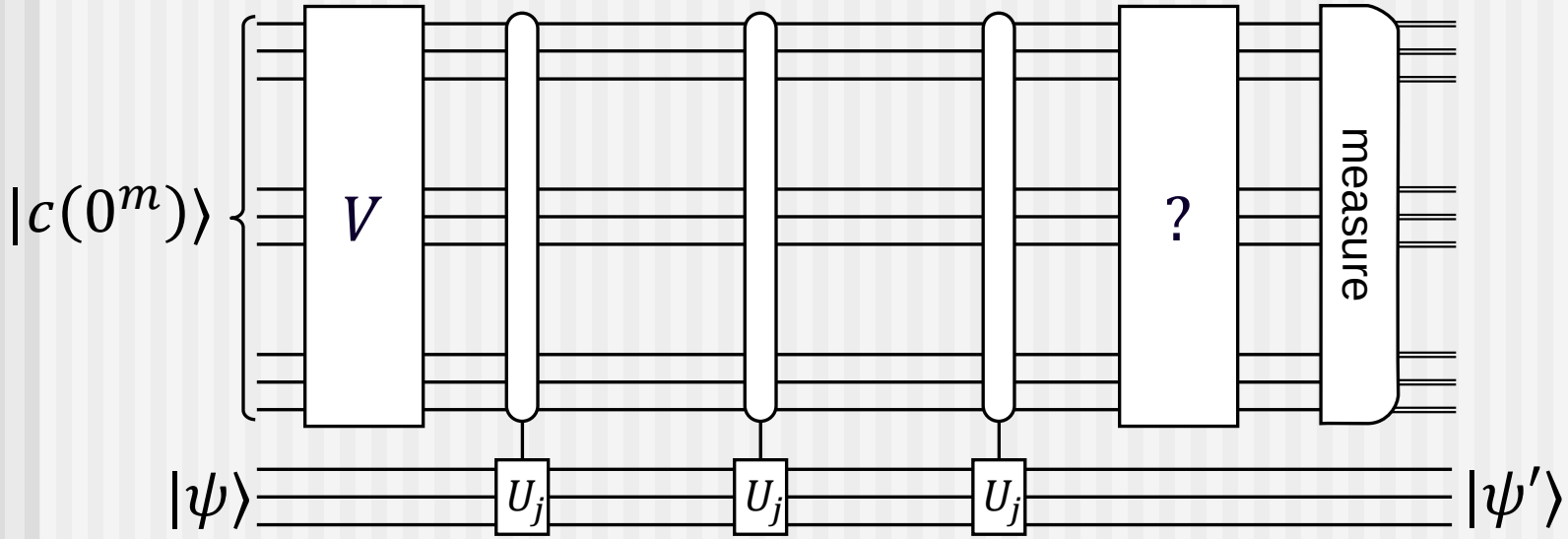
p_2

p_3

Each U_j is U_1 or U_2 depending on whether p_ℓ is even or odd.

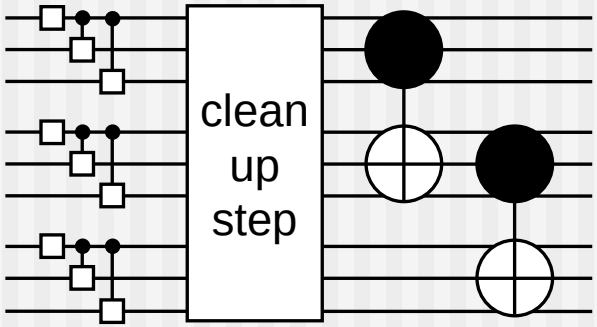
Compressed form: $|c(x)\rangle = |p_1\rangle \ldots |p_{|x|}\rangle |m\rangle^{k-|x|}$

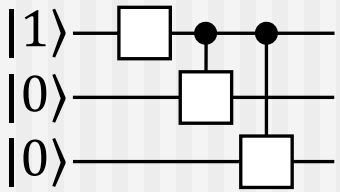
4. Using CGMSY'09 technique



How do we simulate R^m?

$$V =$$





$$\approx \sum_{s=0}^{m-1} \alpha^s \beta |s\rangle + \alpha^m |m\rangle$$

$$V|c(0^m)\rangle \approx \sum_{|x|\leq k} \alpha^{m-|x|} \beta^{|x|} |c(x)\rangle$$

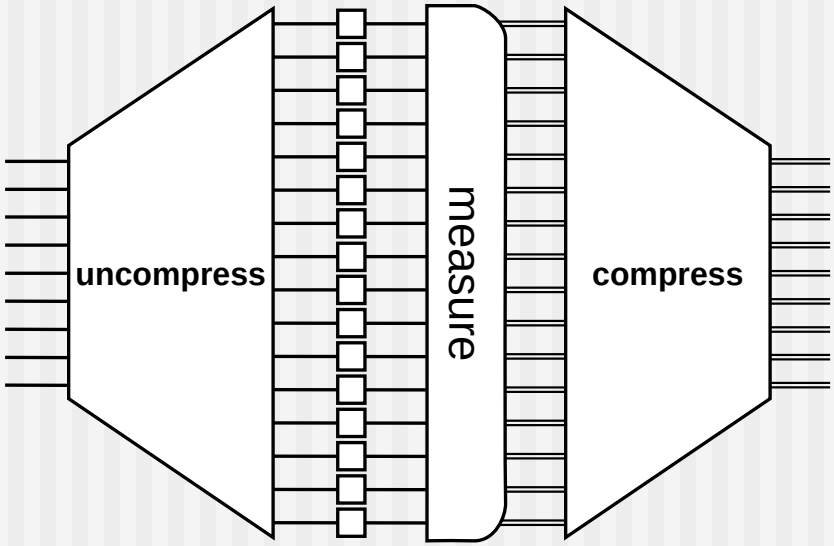
$$|c(x)\rangle = |p_1\rangle \ldots |p_{|x|}\rangle |m\rangle^{k-|x|}$$

D. W. Berry, R. Cleve, and S. Gharibian, arXiv:1211.4637 (2012).

5. Simulation in compressed form

How do we perform the final measurement?

Goal: logically perform this, but without decompressing



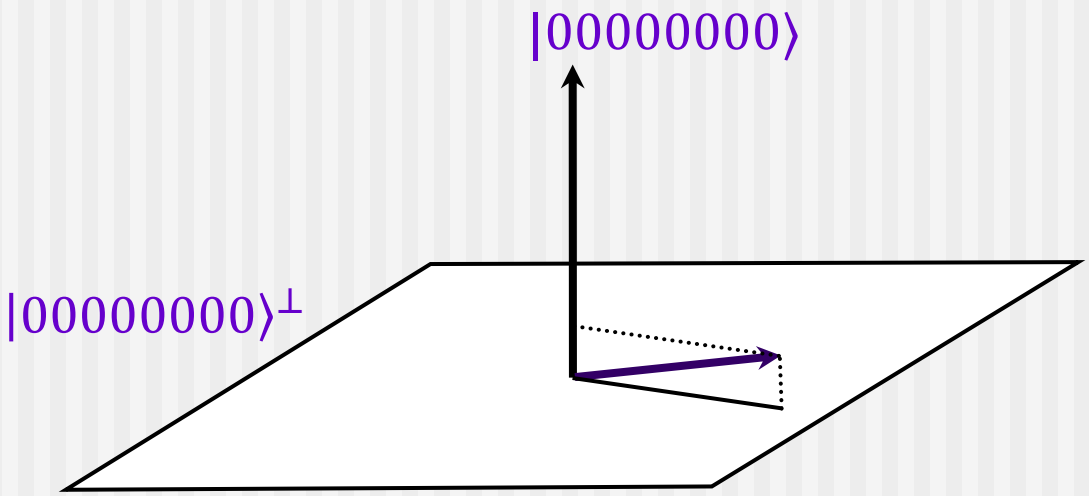
Problem with V is that it ***only*** simulates R^m on state $|c(0^m)\rangle$.

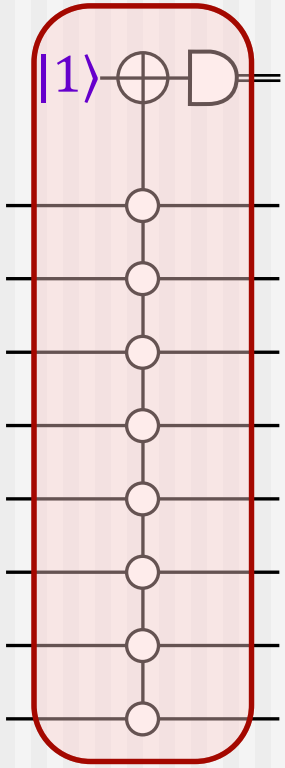
We do get the correct result if we use $V^\dagger$ and happen to get the all-zero measurement result.

D. W. Berry, R. Cleve, and S. Gharibian, arXiv:1211.4637 (2012).

5. Simulation in compressed form

Consider the measurement with outcomes $|00000000\rangle$ and $|00000000\rangle^{\perp}$

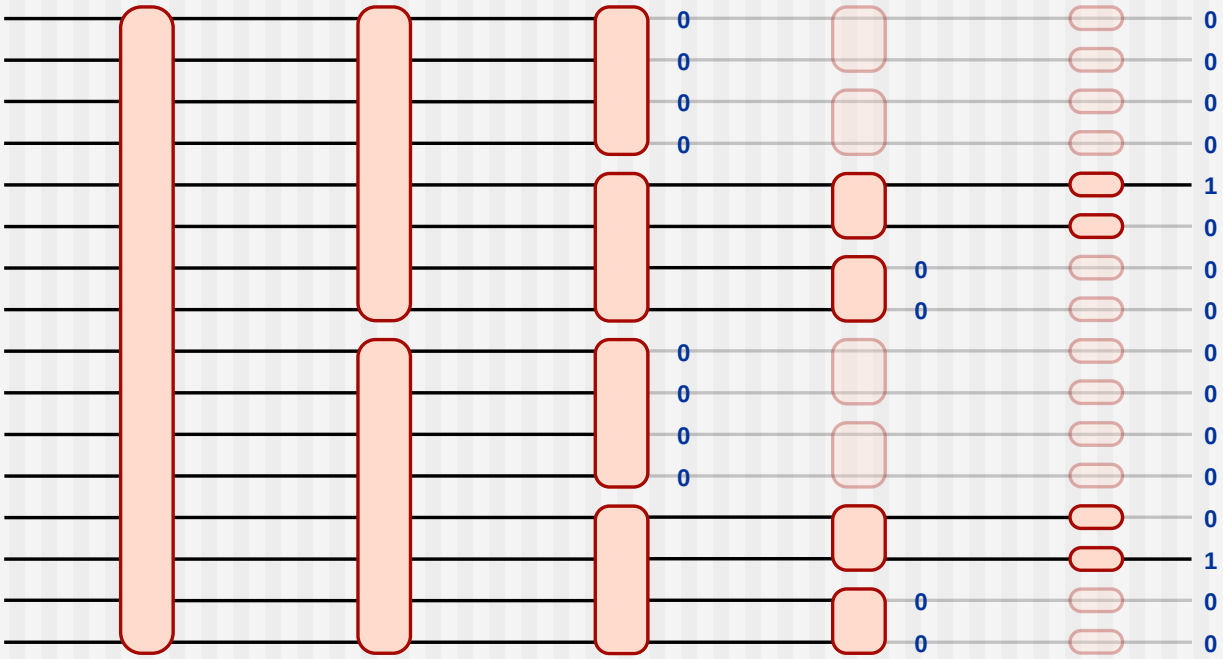




D. W. Berry, R. Cleve, and S. Gharibian, arXiv:1211.4637 (2012).

5. Simulation in compressed form

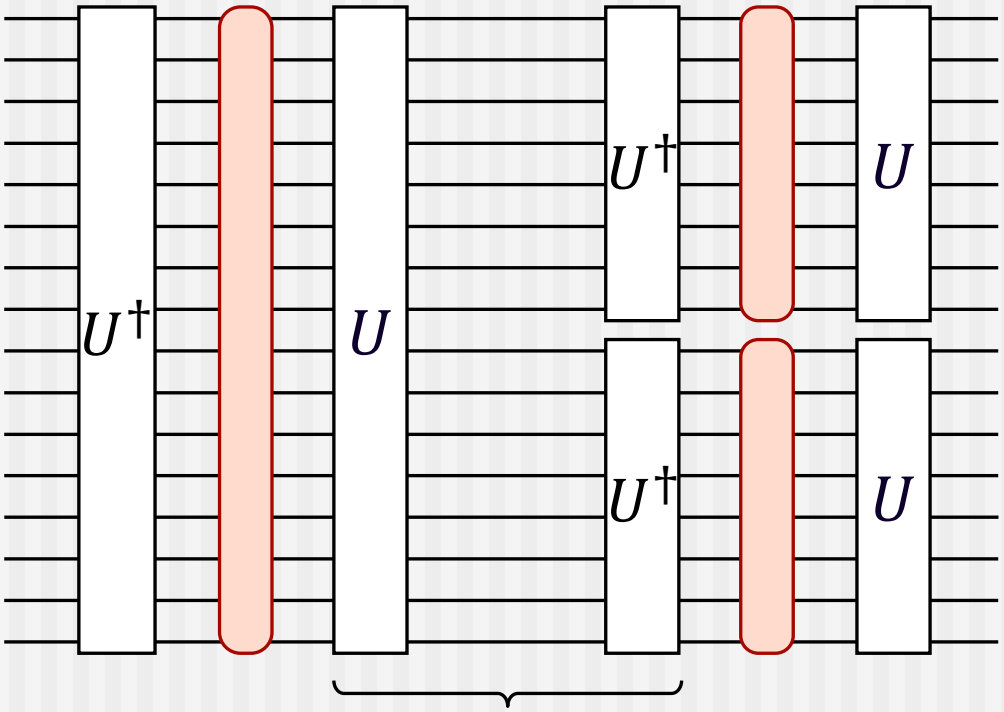
Recursively measure each half (of m qubits):



- Net result: measurement in ***computational basis***.
- If the outcome has Hamming weight k then only $O(k \log m)$ measurement steps are needed.

D. W. Berry, R. Cleve, and S. Gharibian, arXiv:1211.4637 (2012).

5. Simulation in compressed form



this is valid if

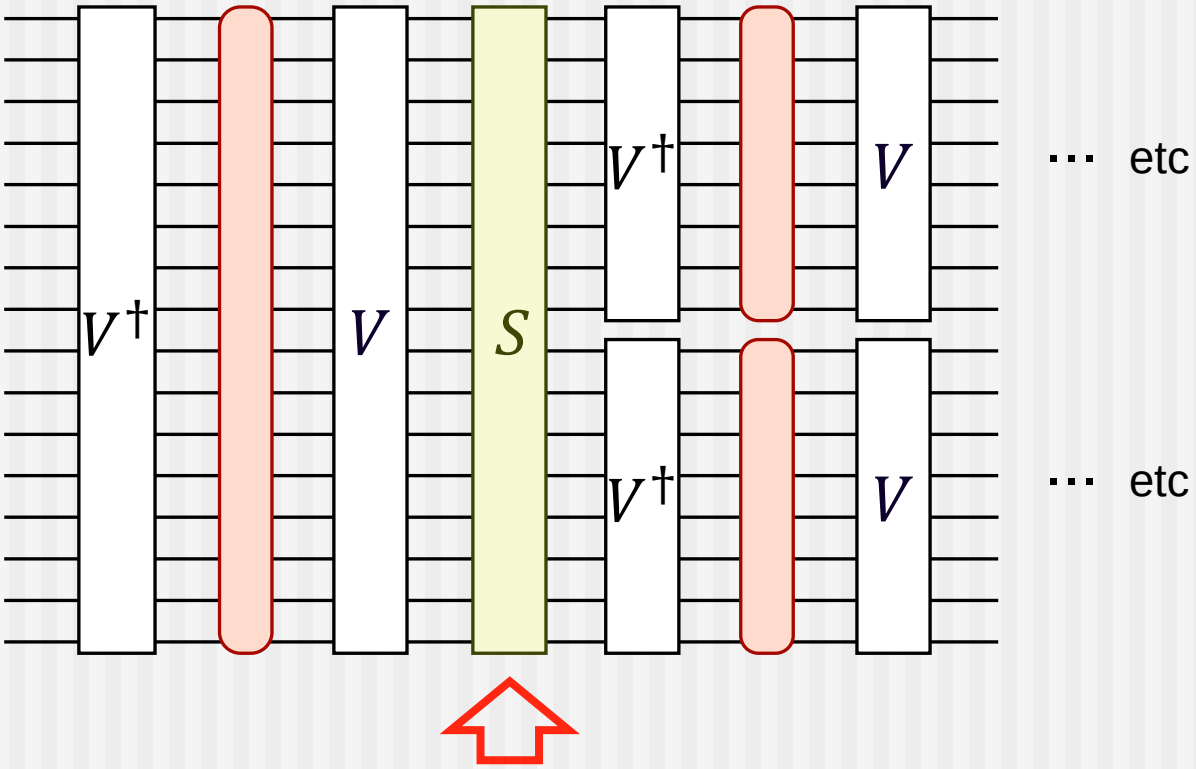
$$U_m = U_{m/2} \otimes U_{m/2}$$

$$U = R^m$$

We are ***only*** using properties of $U_m|0^m\rangle$, $U_{m/2}|0^{m/2}\rangle$, ...

D. W. Berry, R. Cleve, and S. Gharibian, arXiv:1211.4637 (2012).

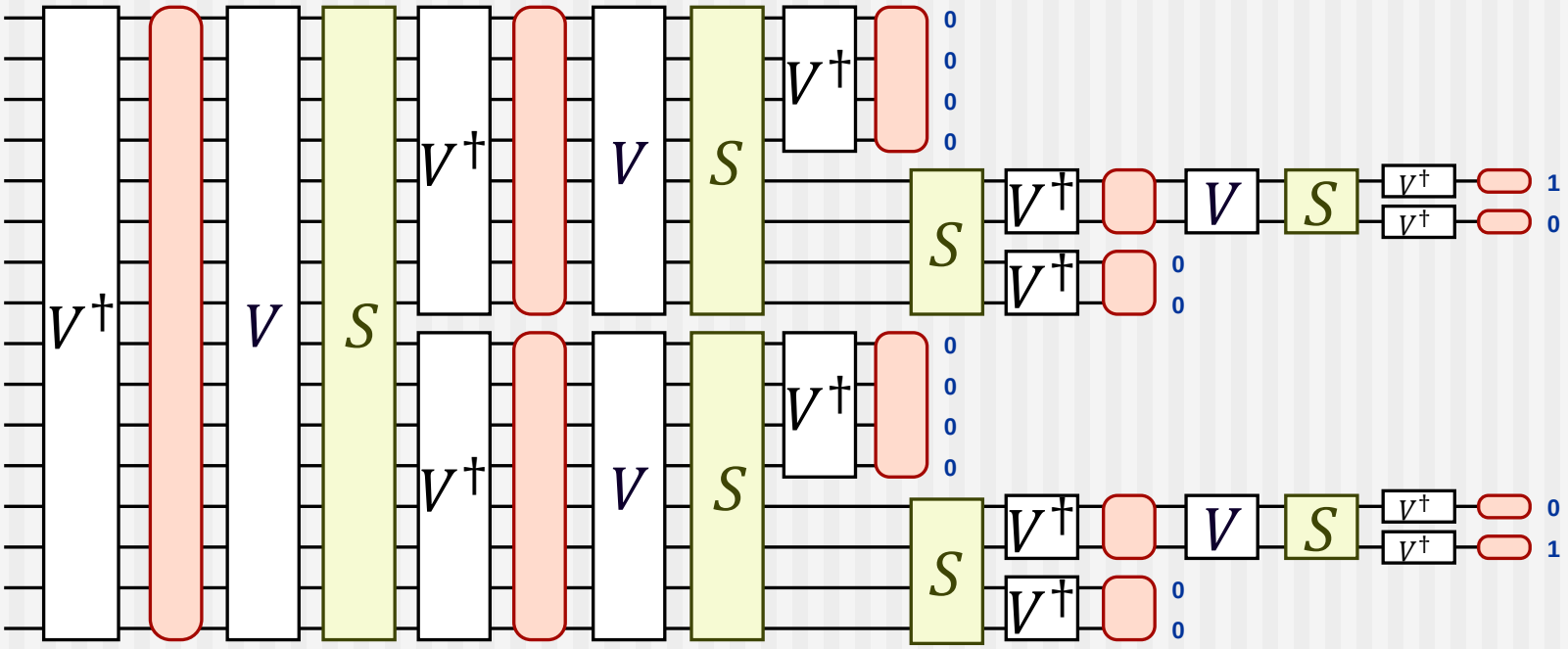
5. Simulation in compressed form



If the states are represented in ***compressed*** form, we need to explicitly ***extract*** left and right halves:

$S|c(x_1x_2)\rangle = |c(x_1)\rangle|c(x_2)\rangle$ is easily implementable.

5. Simulation in compressed form



Bringing it all together

- We divide the Hamiltonian up into $M = O(d^2 \|H\| / \varepsilon_H)$ parts.
- We need to use slightly smaller time segments than $1/4$. We actually subdivide the time into $O(MT\varepsilon_H)$ parts of length $\propto 1/M\varepsilon_H$.
- Each time segment can be simulated with polylog complexity using compression.
- The overall complexity is the number of time segments times a polylog function, and is therefore

$$O(MT\varepsilon_H \times \text{polylog})$$

or

$$O(d^2 \|H\| T \times \text{polylog})$$

Part 2: Linear-time Hamiltonian simulation

How can we perform simulation linear in the evolution time?

How do we beat this?

- Three ingredients:
  1. A Szegedy quantum walk
  2. Coherent phase estimation
  3. Controlled state preparation

- The quantum walk has eigenvalues and eigenvectors related to those for Hamiltonian.
- By using phase estimation, we can estimate the eigenvalue, then implement that actually needed.

A. M. Childs, Commun. Math. Phys. **294**, 581 (2009).
D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).

Szegedy Quantum Walk

- A normal quantum walk has position and coin values

$$|x, c\rangle$$

- The Szegedy quantum walk allows arbitrary dimensions, n and m on the two subsystems.
- A normal quantum walk alternates coin and step operators, e.g.

$$C|x, \pm 1\rangle = (|x, -1\rangle \pm |x, 1\rangle)/\sqrt{2}$$
$$S|x, c\rangle = |x + c, c\rangle$$

- Szegedy quantum walk uses more general controlled “diffusion” operators.

M. Szegedy, quant-ph/0401053 (2004).

Szegedy Quantum Walk

- The “diffusion” operators are of the form

$$2CC^{\dagger} - \mathbb{I}$$
$$2RR^{\dagger} - \mathbb{I}$$

- The first is controlled by the first register and acts on the second register.
- Given some matrix $c[i,j]$, the operator C is defined by

$$|c_i\rangle = \sum_{j=1}^{m} \sqrt{c[i,j]}|j\rangle$$

$$C = \sum_{i=1}^{n} |i\rangle\langle i| \otimes |c_i\rangle$$

M. Szegedy, quant-ph/0401053 (2004).

Szegedy Quantum Walk

- The diffusion operator $2RR^{\dagger} - \mathbb{I}$ is controlled by the second register and acts on the first. Use a similar definition with matrix $r[i,j]$.
- Both are controlled reflections.

- The eigenvalues and eigenvectors of the step of the quantum walk

$$(2CC^{\dagger} - \mathbb{I})(2RR^{\dagger} - \mathbb{I})$$

are related to those of a matrix formed from $c[i,j]$ and $r[i,j]$.

M. Szegedy, quant-ph/0401053 (2004).

Szegedy walk for Hamiltonians

- Basic idea is to use symmetric system, with $n = m$ and
$$c[i,j] = r[i,j] = H_{ij}^*$$
- Then eigenvalues and eigenvectors are related to those of Hamiltonian.

- In reality we need to modify to “lazy” quantum walk, with

$$|c_i\rangle = \sqrt{\frac{\delta}{\|H\|_1}} \sum_{j=1}^{N} \sqrt{H_{ij}^*}\,|j\rangle + \sqrt{1 - \frac{\sigma_i \delta}{\|H\|_1}}\,|N+1\rangle \qquad \sigma_i := \sum_{j=1}^{N} |H_{ij}|$$

laziness parameter

unknown!

A. M. Childs, Commun. Math. Phys. **294**, 581 (2009).

Szegedy walk for Hamiltonians

- Three step process:

  1. Start with state in one of the subsystems, and perform controlled state preparation.

  2. Perform steps of quantum walk to approximate Hamiltonian evolution.

  3. Invert controlled state preparation, so final state is in one of the subsystems.

- Step 2 can just be performed with small δ for lazy quantum walk, or can use phase estimation.

A. M. Childs, Commun. Math. Phys. **294**, 581 (2009).

Simulation via phase estimation

- We can implement one unitary V, but we want to implement a different operation O.
- The operations share eigenstates, but the eigenvalues are related by a function.

$$V|\lambda\rangle = e^{i\lambda}|\lambda\rangle \qquad O|\lambda\rangle = f(\lambda)|\lambda\rangle$$

- One can implement the unitary V many times and perform phase estimation in a coherent way.

$$|\psi\rangle = \sum_\lambda \psi_\lambda |\lambda\rangle \quad \mapsto \sum_\lambda \psi_\lambda |\lambda\rangle|\tilde{\lambda}\rangle \longleftarrow \text{estimate of } \lambda$$

- The eigenvalue $f(\lambda)$ can then be imposed, giving (ignoring error)

$$\sum_\lambda \psi_\lambda f(\lambda)|\lambda\rangle|\tilde{\lambda}\rangle$$

- Inverting the phase estimation gives

$$\sum_\lambda \psi_\lambda f(\lambda)|\lambda\rangle = O|\psi\rangle$$

Szegedy Walk for Hamiltonians

- A Hamiltonian has eigenvalues μ.
- V is the step of a quantum walk, and has eigenvalues
$$e^{i\lambda} = \pm e^{\pm i \arcsin \mu\delta}$$
- The operation O is evolution under the Hamiltonian. It has eigenvalues
$$e^{-i\mu T}$$

A. M. Childs, Commun. Math. Phys. **294**, 581 (2009).

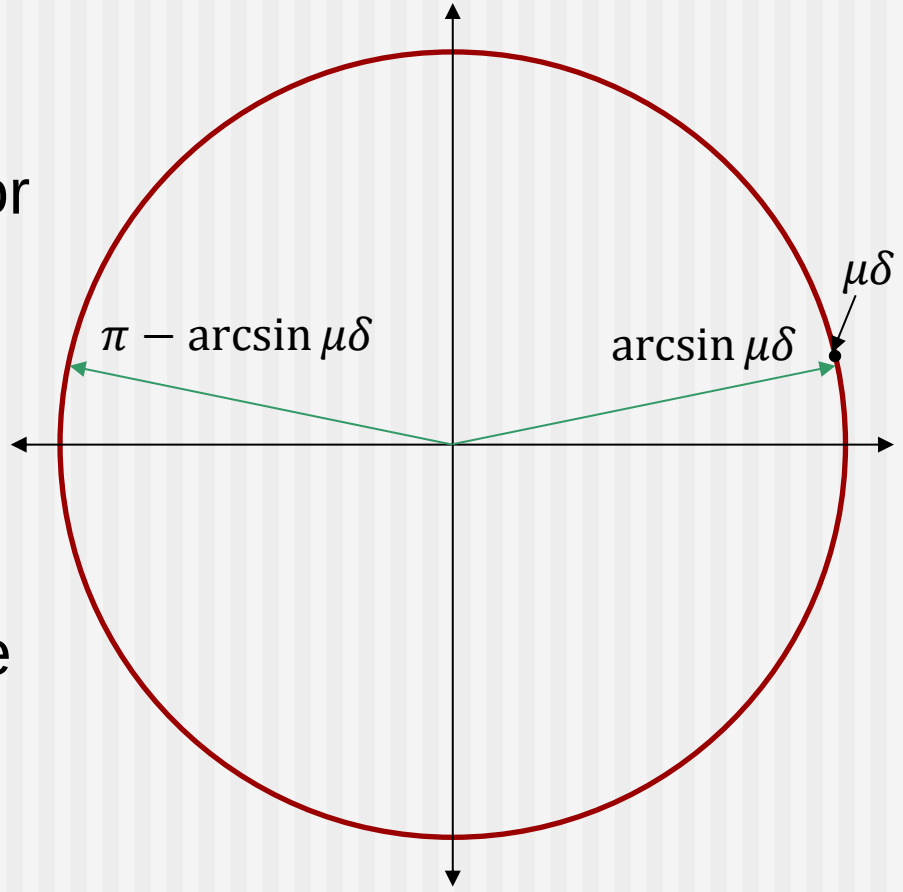
New Advances

1. Combine the lazy quantum walk with the phase estimation approach to improve efficiency.
2. Explicit procedure to achieve controlled state preparation for steps of quantum walk.
3. Combine with Trotterisation approach.
4. Apply to implement unitary operations.

D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).

Advance #1

- Step of quantum walk has eigenvalues
$$\pm e^{\pm i \arcsin \mu\delta}$$
- For small δ, arcsin is close to linear.
- Lazy quantum walk yields error proportional to error in nonlinearity.
- Also use phase estimation to estimate μ. We thereby estimate the error due to the nonlinearity.
- Using this to correct the phase yields a result more accurate than the lazy quantum walk or phase estimation alone.



D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).

Advance #2

- State preparation by amplitude amplification was invented by Grover (2000).
- Start with equal superposition state and ancilla

$$\frac{1}{\sqrt{N}}\sum_{k=1}^{N}|k\rangle|0\rangle$$

- Rotate ancilla according to amplitude for state to be prepared

$$|\psi^b\rangle = \frac{1}{\sqrt{N}}\sum_{k=1}^{N}|k\rangle\left(\psi_k|0\rangle + \sqrt{1-|\psi_k|^2}|1\rangle\right)$$

- Amplitude amplification yields component where ancilla is zero.

D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).
L. K. Grover, PRL **85**, 1334 (2000).

Advance #2

- Compare initial state for Grover preparation technique

$$|\psi^b\rangle = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} |k\rangle \left(\psi_k |0\rangle + \sqrt{1 - |\psi_k|^2} |1\rangle \right)$$

to that we wish to prepare

$$|c_i\rangle = \sqrt{\frac{\delta}{\|H\|_1}} \sum_{j=1}^{N} \sqrt{H_{ij}^*} |j\rangle + \sqrt{1 - \frac{\sigma_i \delta}{\|H\|_1}} |N+1\rangle$$

- Crucial insight is that second term for Grover preparation may take the role of laziness term.
- This means we do not need to rotate all the way to the final state – or know σ_i.

D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).

Results from #1 and #2

Best case:

$\|H\|_{\max}\|H\|_1 \propto \|H\|^2$

then

$\propto \sqrt{d}$

the

axim

Worst case:

$\|H\|_1 \propto \sqrt{D}\|H\|$

then

$\propto d^{3/4}$

- An alternative bound suitable for non-sparse cases is

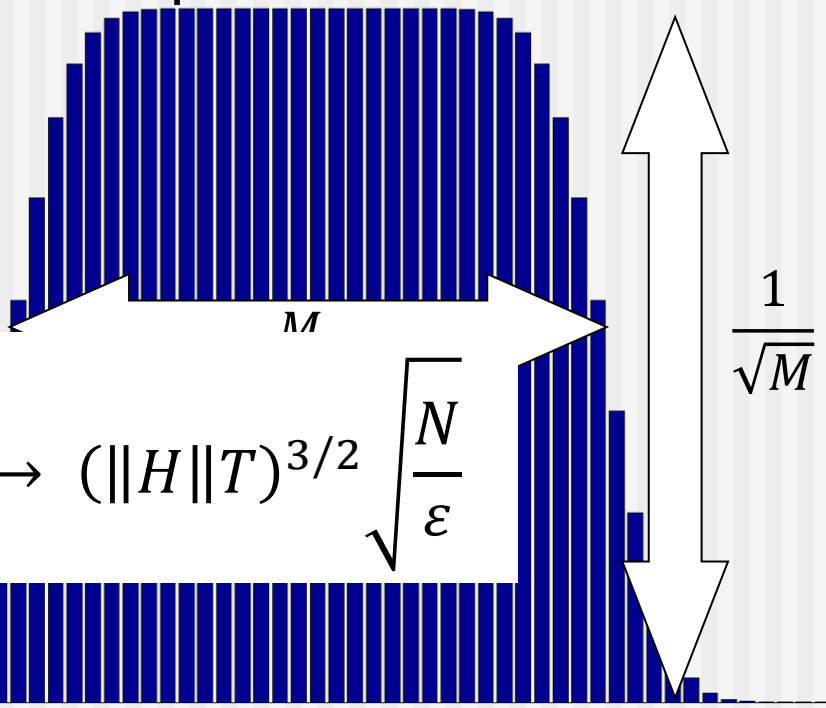
$$T^{3/2}\sqrt{\frac{d\|H\|_{\max}\|H\|_1\|H\|}{\varepsilon}}$$

D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).

Advance #3

- Consider the non-sparse case – we can replace *d* with N, and want to improve scaling in N rather than T.
- Using first two advances enables us to obtain scaling close to $N^{1/2}$ (the fundamental limit) in many cases.
- The cases where this works are those where the non-zero elements are of comparable size.

$$\|H\|_{\max}\|H\|_1 \propto \|H\|^2$$



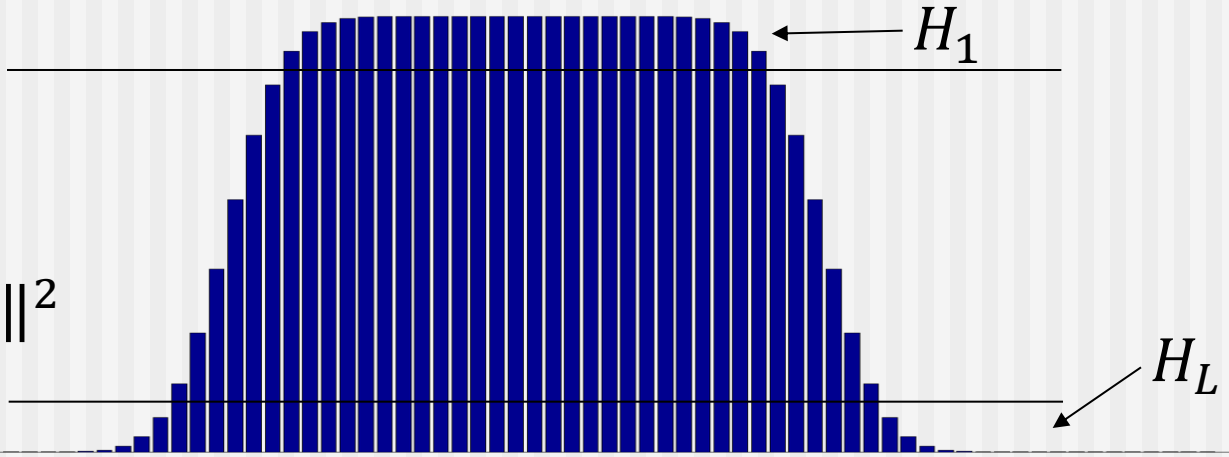
$$T^{3/2}\sqrt{\frac{d\|H\|_{\max}\|H\|_1\|H\|}{\varepsilon}} \rightarrow (\|H\|T)^{3/2}\sqrt{\frac{N}{\varepsilon}}$$

Advance #3

- Divide up the Hamiltonian into a sum of terms, each of which has non-zero elements with similar magnitude.
- Each of the individual Hamiltonians is simulated via the quantum walk approach.
- By using Trotterisation we recombine these simulations to give us simulation of the overall Hamiltonian.
- The complexity scales as $N^{1/2}$ provided the norms of the components do not blow up.

$$H = \sum_{\ell=1}^{L} H_\ell$$

$$\|H_\ell\|_{\max} \|H_\ell\|_1 \propto \|H_\ell\|^2$$



Advance #3

- If we allow the possibility that the norms are not well-behaved, then the analysis becomes much more complicated.

- We must use a host of tricks to deal with this case:
  1. Find a bound on the norms, with a special trick for the Hamiltonian with the smallest matrix elements (H_L).
  2. Use preparation without amplitude amplification for H_L.
  3. Choose a special sequence of cut-offs that varies double-exponentially.
  4. Use nested Trotter-Suzuki formulae, combining all Hamiltonians except that with largest matrix elements (H_1) first, then combining those with H_1 next.

D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).

Results from #1 to #3

- Let $\Upsilon(H)$ be the largest ratio that the norm of H can be increased by breaking it up according to the component magnitude.
- The complexity is no more than

$$(\|H\|T)^{3/2}(\log N)^{7/4}\sqrt{\frac{N\Upsilon(H)}{\varepsilon}}$$

- Method for pathological cases gives complexity bound

$$[(\log\log N)\|H\|T]^{4/3}N^{2/3}\varepsilon^{-1/3}$$

D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).

Advance #4

- A closely related problem is how to implement a unitary.
- Method is (Jordan & Wocjan 2009; Childs) construct a Hamiltonian from unitary as

$$H = \begin{bmatrix} 0 & U \\ U^\dagger & 0 \end{bmatrix}$$

- Now simulate evolution under this Hamiltonian

$$e^{-iHT} = \mathbb{I} \cos T - iH \sin T$$

- Simulating for time $t = \pi/2$ gives

$$e^{-iH\pi 2}|1\rangle|\psi\rangle = -iH|1\rangle|\psi\rangle = -i|0\rangle U|\psi\rangle$$

D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).
S. P. Jordan & P. Wocjan, Phys. Rev. A **80**, 062301 (2009).

Advance #4

- We can use

$$\|H\| = \|U\| = 1, \qquad t = \pi/2$$

$$\|H\|_1 = \max\left(\|U\|_1, \left\|U^\dagger\right\|_1\right), \qquad \|H\|_{\max} = \|U\|_{\max}$$

- The results from before directly give

$$(\log N)^{7/4}\sqrt{\frac{N\Upsilon(U)}{\varepsilon}}$$

$$(\log\log N)^{4/3} N^{2/3} \varepsilon^{-1/3}$$

D. W. Berry and A. M. Childs, Quantum Information and Computation **12**, 29 (2012).

Search Problem

- We can encode a search problem in a unitary.

$$U = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad U_{ij} = f(i-j), \quad f(1) = 1$$

- We have $d = N$, because we do not know in advance where the non-zero elements are.
- Multiplying gives vector revealing location of non-zero element.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Conclusions

- We have complexity of sparse Hamiltonian simulation scaling as

$$O(d^2 \|H\| T \times \text{polylog})$$

- This provides the polylog scaling in both the error and the derivative of the Hamiltonian.

- A quantum walk approach gives us scaling as the minimum of

$$\frac{d\|H\|T}{\sqrt{\varepsilon}} \qquad \frac{(\|H\|T)^{4/3} d^{2/3}}{\varepsilon^{1/3}}$$

- This provides strictly linear scaling in $\|H\|T$, as well as improved scaling in the sparseness.