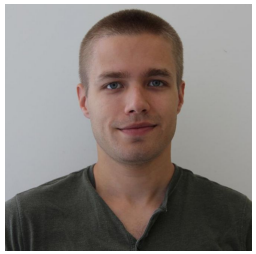
Faster Digital Quantum Simulation by Randomization

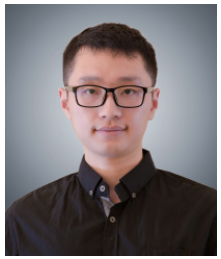
Yuan Su
University of Maryland

Dominic Berry
Macquarie

Andrew Childs
Maryland



Aaron Ostrander
Maryland



Xin Wang
Maryland
Baidu

Nathan Wiebe
Google
Washington

---

[1] Faster quantum simulation by randomization, arXiv:1805.08385.

[2] Time-dependent Hamiltonian simulation with L^1-norm scaling, arXiv:1906.07115.

Outline

Outline

Quantum computing basics

Quantum operation

Quantum operations on a closed quantum system are described by unitaries

$$U^\dagger U = UU^\dagger = I.$$

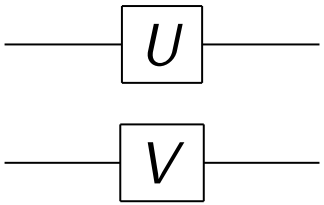
- Single-qubit operations are unitaries acting on $\mathbb{C}^2$:

$$\boxed{\sigma_x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \boxed{\sigma_y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \boxed{\sigma_z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

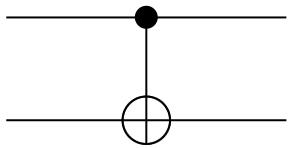
$$\boxed{\text{Had}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \boxed{\text{P}} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad \boxed{\text{T}} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}.$$

Quantum computing basics

- n-qubit operations are unitaries acting on $\mathbb{C}^{2^n}$:



$= U \otimes V,$



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

- In the circuit model, quantum computation is realized by a sequence of elementary quantum gates (such as CNOT, Had, P, and T).

- Complexity is quantified by the number of gates used in the computation.

Factoring and unstructured search

- The difficulty of integer factoring has been used to design modern cryptosystems.
- In 1994, Peter Shor designed a polynomial time algorithm for factoring integers, based on a quantum algorithm for computing periodicity.

- Grover gave a quantum algorithm that solves an unstructured search problem with n possible solutions, using only $O(\sqrt{n})$ queries.
- This is a polynomial speedup to a very generic problem.

Quantum algorithms

- Algebraic problems:
  Factoring integers, computing discrete logarithms, ...
- Simulating quantum dynamics:
  Quantum chemistry, materials science, ...
- Matrix arithmetics:
  Linear equations, differential equations, ...
- Unstructured search:
  Collision finding, graph problems, ...
- ...

Quantum Algorithm Zoo, `https://quantumalgorithmzoo.org/`

Quantum algorithms

- Algebraic problems:
  Factoring integers, computing discrete logarithms, ...
- **Simulating quantum dynamics:**
  Quantum chemistry, materials science, ...
- Matrix arithmetics:
  Linear equations, differential equations, ...
- Unstructured search:
  Collision finding, graph problems, ...
- ...

Quantum Algorithm Zoo, `https://quantumalgorithmzoo.org/`

Outline

Quantum dynamics

- The dynamics of a quantum system are determined by its Hamiltonian $H(t)$ according to the Schrödinger equation

$$\frac{\mathrm{d}}{\mathrm{d}t}U(t) = -iH(t)U(t), \qquad U(0) = I.$$

- $H(t)^\dagger = H(t)$ is Hermitian, so $U(t)$ is unitary.

- When $H(t) \equiv H$ is time-independent, we have the closed-form solution $U(t) = e^{-itH}$.

- For a general $H(t)$, we formally represent the evolution as $U(t) = \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}\tau\ H(\tau)\right)$.

Digital quantum simulation

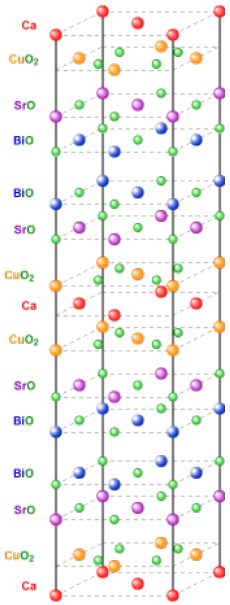
Quantum simulation problem

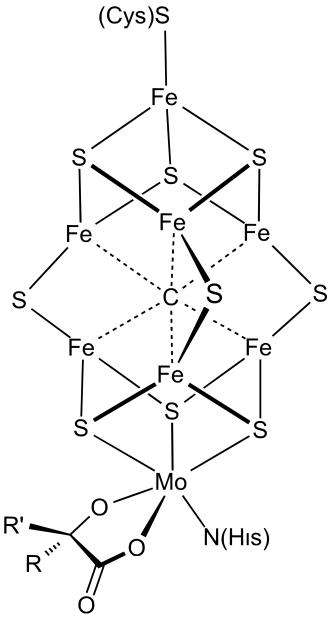
Given a description of the Hamiltonian H and evolution time t, perform e^{-itH} up to some error ϵ (in spectral norm)

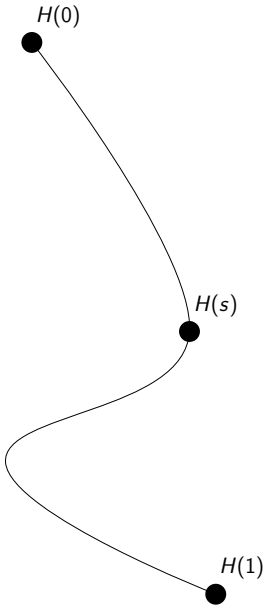
$$\left\| U - e^{-itH} \right\| \leq \epsilon.$$

- A quantum computer can simulate the Hamiltonian efficiently if H is local/sparse.
- Upon measurement, it can efficiently answer questions that a classical one cannot.
- More generally, we can consider simulating time-dependent Hamiltonians.

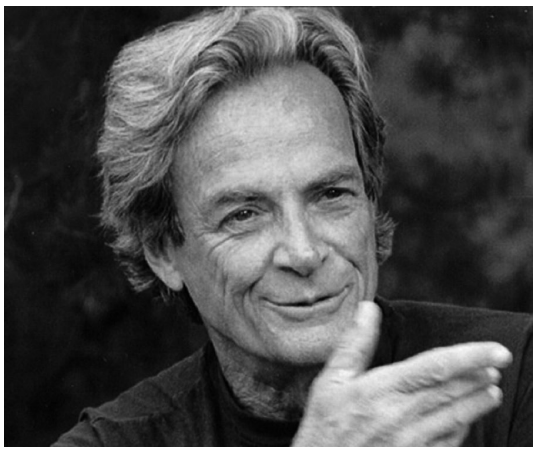
Reasons to study quantum simulation







$$A \quad x = b$$



"...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

Product formulas

- Also known as Trotterization or the splitting method.
- Target system: $H = \sum_{j=1}^{n} h_j H_j$, where each H_j is both unitary and Hermitian and can be implemented with constant cost.
- Can use the first-order Lie-Trotter formula[1]

$$S_1(t) := e^{-ith_nH_n} \cdots e^{-ith_1H_1} \approx e^{-it(h_nH_n+\cdots+h_1H_1)}.$$

- Generalizations to $(2k)$th-order formula $S_{2k}(t)$ exist[2].

[1] [Lloyd 96]
[2] [Suzuki 92]

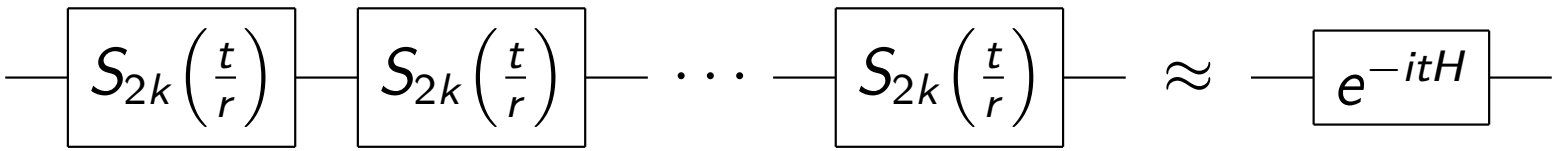
The product-formula algorithm

The product-formula algorithm

- Divide the evolution into r steps.
- Use the first-/$(2k)$th-order product formula within each step.
- Choose r sufficiently large so that

$$\left\| S_{2k}^r\left(\frac{t}{r}\right) - e^{-itH} \right\| \leq \epsilon.$$

- Circuit representation:



Complexity of product-formula simulation

- Within a short time t/r, we have **Trotter error**

$$\left\| S_{2k}\left(\frac{t}{r}\right) - e^{-i\frac{t}{r}H} \right\| = O\left(\left(\frac{\|h\|_1 t}{r}\right)^{2k+1}\right).$$

- Repeating r **Trotter steps**, the total error becomes

$$\left\| S_{2k}^r\left(\frac{t}{r}\right) - e^{-itH} \right\| = O\left(\frac{(\|h\|_1 t)^{2k+1}}{r^{2k}}\right).$$

- To achieve accuracy ϵ, it suffices to choose **Trotter number**

$$r = O\left(\frac{(\|h\|_1 t)^{1+\frac{1}{2k}}}{\epsilon^{\frac{1}{2k}}}\right) = O\left(\frac{(nt)^{1+\frac{1}{2k}}}{\epsilon^{\frac{1}{2k}}}\right),$$

giving gate complexity $O(n^{2+1/2k} t^{1+1/2k} / \epsilon^{1/2k})$.

Other simulation algorithms

- Recent algorithms have improved asymptotic performance as a function of t and ϵ over the product-formula approach ...

Algorithm	Gate complexity (n, t)
Product formula (PF), 1st order	$O(n^3 t^2)$
Product formula (PF), $(2k)$th order	$O(5^{2k} n^{2+1/2k} t^{1+1/2k})$
Quantum walk	$O(n^3 t \log n)$
Fractional-query simulation	$O\left(n^3 t \frac{\log t}{\log \log t}\right)$
Taylor series[3](TS)	$O\left(n^2 t \frac{\log n \log t}{\log \log t}\right)$
Linear combination of quantum walk	$O\left(n^3 t \frac{\log t}{\log \log t}\right)$
Quantum signal processing[4](QSP)	$O(n^2 t \log n)$

- **... but the n-scaling seems unavoidable.**

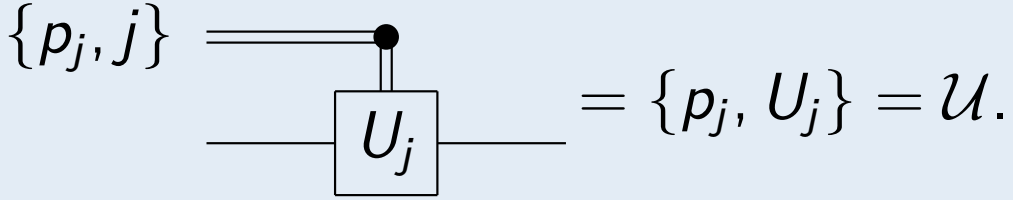
[3] [Berry, Childs, Cleve, Kothari, Somma 15]
[4] [Low, Chuang 17]

Outline

Mixed-unitary quantum channel

Mixed-unitary quantum channel

Performing unitary operation U_j with probability p_j implements the mixed-unitary quantum channel $\mathcal{U} = \{p_j, U_j\}$



- A unitary operation U can be seen as a unitary channel with $\mathcal{U} = \{1, U\}$.

- For a fair comparison, we may describe all quantum simulation algorithms using quantum channels and fix a channel distance as the unified error metric.

Diamond-norm distance

- The distance between quantum channels $\mathcal{U}$ and $\mathcal{V}$ can be quantified by the diamond-norm distance $\|\mathcal{U}-\mathcal{V}\|_\diamond$.

- If unitary operations U and V are close in spectral-norm distance, the corresponding channels $\mathcal{U}$ and $\mathcal{V}$ are close in diamond-norm distance.

- More specifically[5],

$$\|\mathcal{U}-\mathcal{V}\|_\diamond \leq 2\|U-V\|.$$

[5] [Berry, Childs, Kothari 15]

Mixing lemma

Mixing lemma[6]

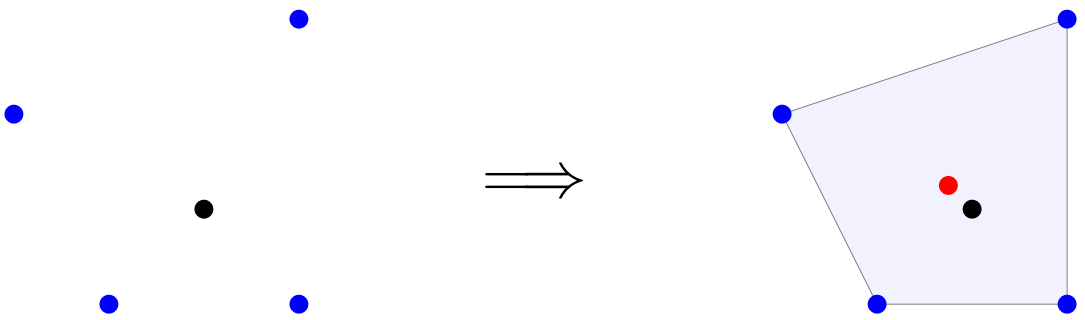
Given mixed-unitary channel $\mathcal{U} = \{p_j, U_j\}$ and unitary channel $\mathcal{V} = \{1, V\}$, suppose that

(i) $\|U_j - V\| \leq a$ for all j; and

(ii) $\left\|\left(\sum_j p_j U_j\right) - V\right\| \leq b$.

Then, $\|\mathcal{U} - \mathcal{V}\|_\diamond \leq a^2 + 2b$.

- Geometrical interpretation:



[6] [Campbell 17], [Hastings 17]

Outline

Randomized first-order algorithm

- With $H = A + B$ and $\lambda = -it$,

$$S_{1,\text{orig}}(\lambda) := e^{\lambda B} e^{\lambda A} = I + \lambda(A+B) + \frac{\lambda^2}{2}(A^2 + 2AB + B^2) + O(\lambda^3),$$
$$V(\lambda) := e^{\lambda(A+B)} = I + \lambda(A+B) + \frac{\lambda^2}{2}(A^2 + AB + BA + B^2) + O(\lambda^3).$$

- Using the triangle inequality,

$$\|S_{1,\text{orig}}(\lambda) - V(\lambda)\| \leq \frac{|\lambda|^2}{2} \|[A, B]\| + O\left(\lambda^3\right).$$

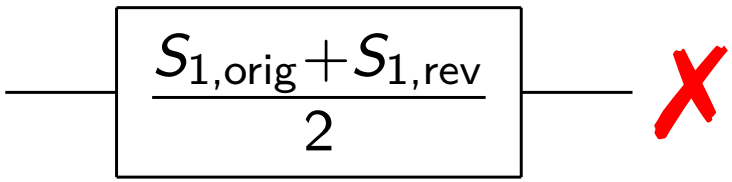
- A uniform mixture of $S_{1,\text{orig}}(\lambda)$ and $S_{1,\text{rev}}(\lambda) := e^{\lambda A} e^{\lambda B}$ gives a better approximation[7]

$$\left\| \frac{1}{2}\left(S_{1,\text{orig}}(\lambda) + S_{1,\text{rev}}(\lambda)\right) - V(\lambda) \right\| = O\left(\lambda^3\right).$$

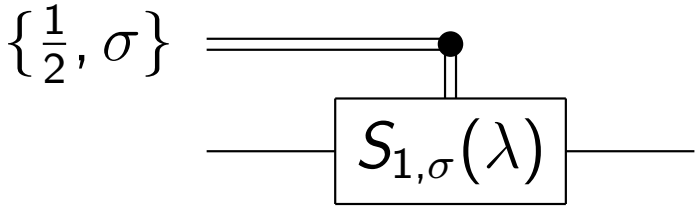
[7] [Zhang 12]

Implementing randomized simulation

- $\left(S_{1,\mathrm{orig}}(\lambda) + S_{1,\mathrm{rev}}(\lambda)\right)/2$ cannot be operationally implemented on a quantum computer.



- **Workaround:** apply $S_{1,\sigma}(\lambda)$ with $\sigma \in \{\mathrm{orig}, \mathrm{rev}\}$ chosen uniformly at random to implement a quantum channel.

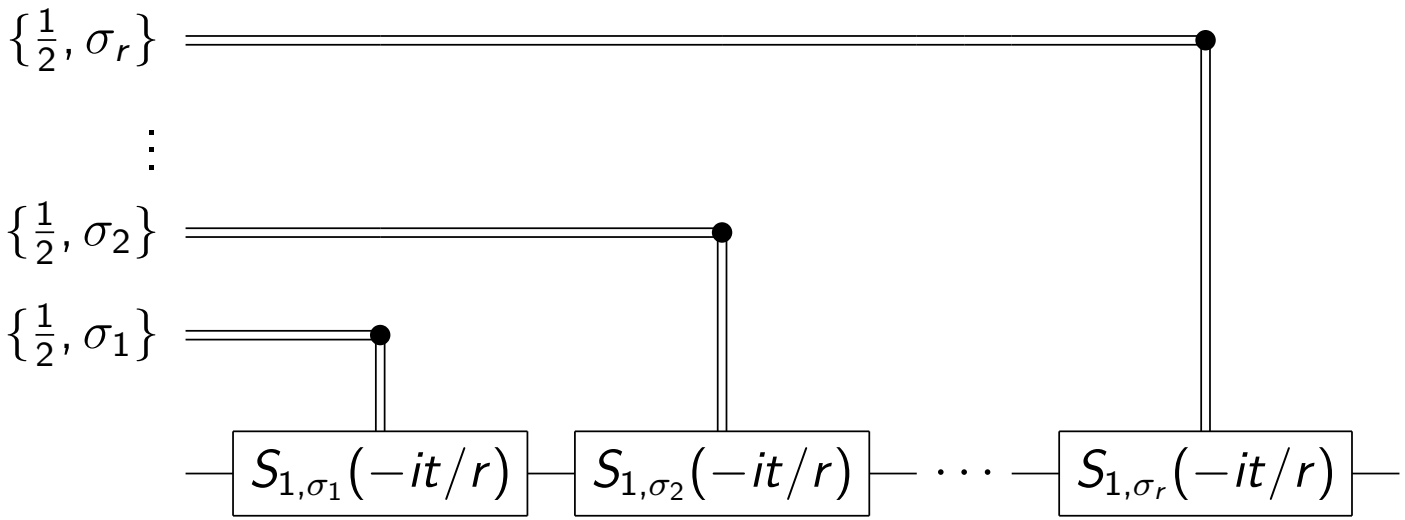


Complexity analysis

- Invoking the mixing lemma with $a = O(\lambda^2)$ and $b = O(\lambda^3)$,

$$\left\| \tfrac{1}{2}\left(S_{1,\mathrm{orig}}(\lambda) + S_{1,\mathrm{rev}}(\lambda)\right) - V(\lambda) \right\|_\diamond = O\left(\lambda^3\right).$$

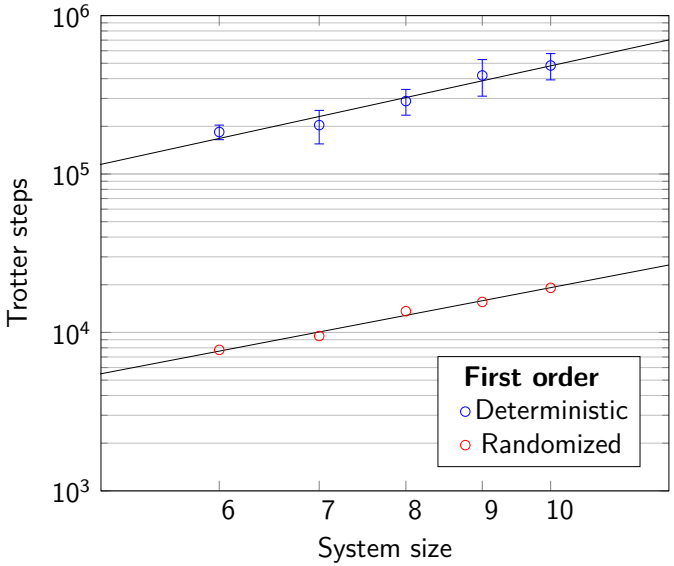
- We can thus **use randomness to promote a first-order formula to a second-order formula.**
- To simulate for time t, divide the evolution into r steps and use the randomized product formula within each step.

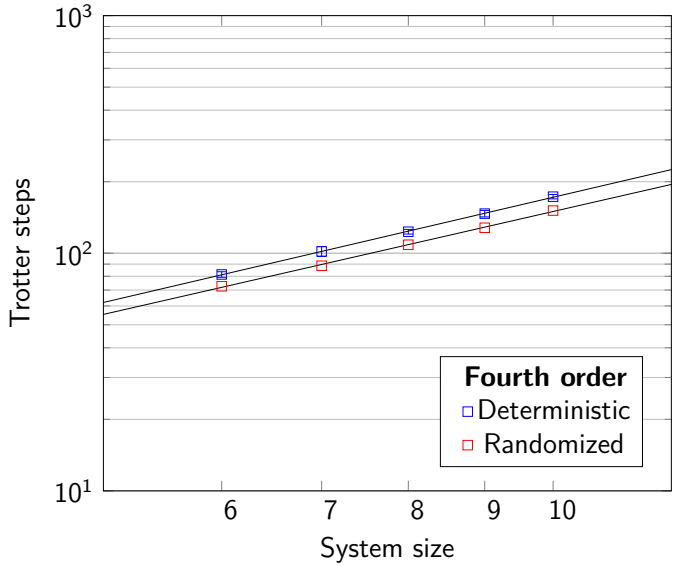


Randomized higher-order algorithm

- The analysis for higher-order formulas is more challenging due to their more complicated combinatorial structures.
- Introduce **more randomness:** randomly permute all n summands instead of choosing whether or not to reverse.

Bounds on r	1st-order	$(2k)$th-order
Deterministic	$O\left(\frac{(nt)^2}{\epsilon}\right)$	$O\left(nt\left(\frac{nt}{\epsilon}\right)^{\frac{1}{2k}}\right)$
Randomized	$O\left(nt\left(\frac{nt}{\epsilon}\right)^{\frac{1}{2}}\right)$	$\max\left\{O\left(nt\left(\frac{nt}{\epsilon}\right)^{\frac{1}{4k+1}}\right), O\left(nt\left(\frac{t}{\epsilon}\right)^{\frac{1}{2k}}\right)\right\}$





Outline

Simulation without explicit n-scaling

- Target system: $H = \sum_{j=1}^{n} h_j H_j$, where each H_j is both unitary and Hermitian and can be implemented with constant cost.

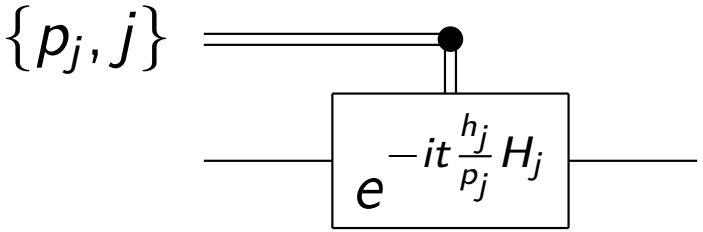
Algorithm	Gate complexity (n, t)
Product formula (PF), 1st order	$O(n^3 t^2)$
Product formula (PF), $(2k)$th order	$O(5^{2k} n^{2+1/2k} t^{1+1/2k})$
Quantum walk	$O(n^3 t \log n)$
Fractional-query simulation	$O\left(n^3 t \frac{\log t}{\log \log t}\right)$
Taylor series (TS)	$O\left(n^2 t \frac{\log n \log t}{\log \log t}\right)$
Linear combination of quantum walk	$O\left(n^3 t \frac{\log t}{\log \log t}\right)$
Quantum signal processing (QSP)	$O(n^2 t \log n)$

- A deterministic approach has complexity scaling at least linearly in n. Not optimal for large n and unbalanced $|h_j|$.

A quantum drift protocol

- To reduce the simulation cost, we bias toward those terms with large coefficients.

- More specifically[8], we choose the jth term with probability $p_j = |h_j| / \|h\|_1$ and perform $e^{-ith_jH_j/p_j}$.

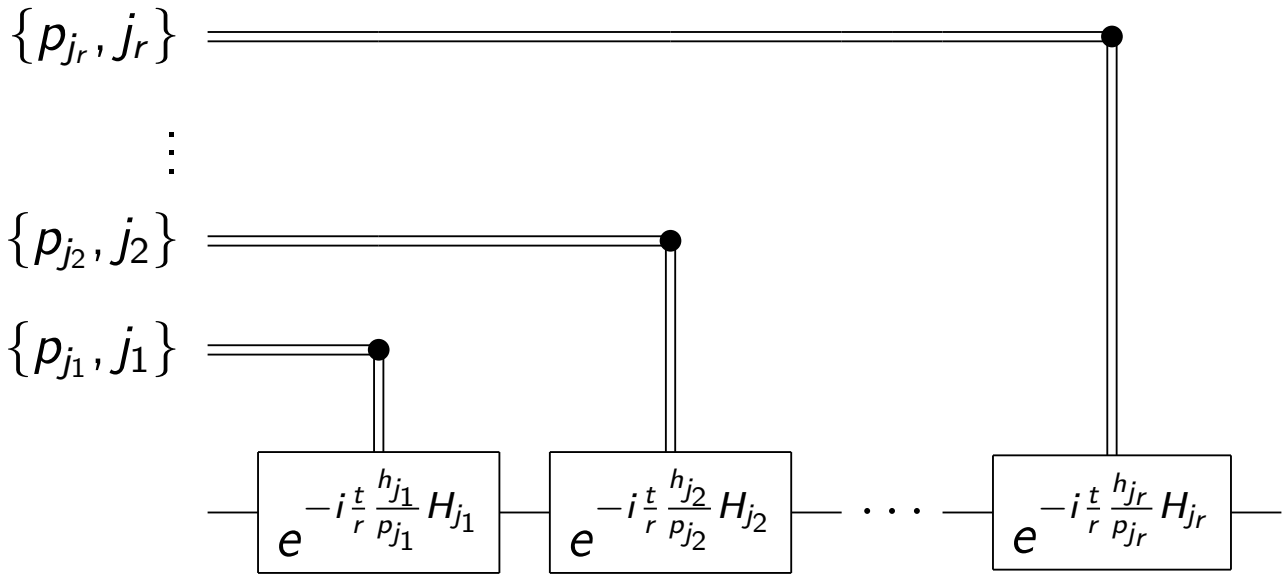


- Applying the mixing lemma with $a = \left\| e^{-ith_jH_j/p_j} - e^{-itH} \right\| = O(t)$ and $b = \left\| \sum_j p_j e^{-ith_jH_j/p_j} - e^{-itH} \right\| = O(t^2)$, we see that the protocol is first-order accurate.

[8] [Campbell 19]

Complexity analysis

- A more careful calculation shows that $\{p_j, e^{-ith_j H_j/p_j}\}$ approximates $\{1, e^{-itH}\}$ with diamond-norm error $4(\|h\|_1 t)^2$.
- We can thus **use randomness to avoid the n-scaling of quantum simulation.**
- To simulate for time t, divide the evolution into r steps and use the quantum drift protocol within each step.

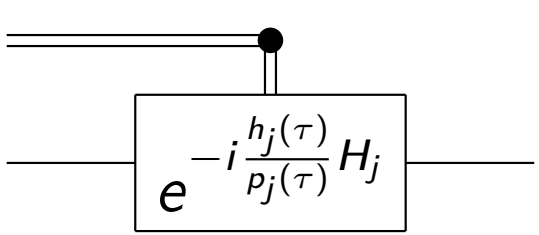


Simulation without L^∞-norm scaling

- Target system: $H(t) = \sum_{j=1}^{n} h_j(t) H_j$, where H_j are both unitary and Hermitian and can be implemented with constant cost.
- Reduces to the time-independent case for each short-time step.
- Intuitively the complexity should scale with the L^1 norm $\int_0^t \mathrm{d}\tau \, \|h(\tau)\|_1$. But previous simulation algorithms scale with the L^∞ norm $\sup_{\tau\in[0,t]} \max_{j\in\{1,\ldots,n\}} \|h_j(\tau)\|$.
- Not optimal when the Hamiltonian varies significantly with time.

A continuous quantum drift protocol

- We sample j and τ according to the probability density function $p_j(\tau) = |h_j(\tau)| / \int_0^t \mathrm{d}\tau \, \|h(\tau)\|_1$ and perform $e^{-ih_j(\tau)H_j/p_j(\tau)}$.



- Applying the mixing lemma with

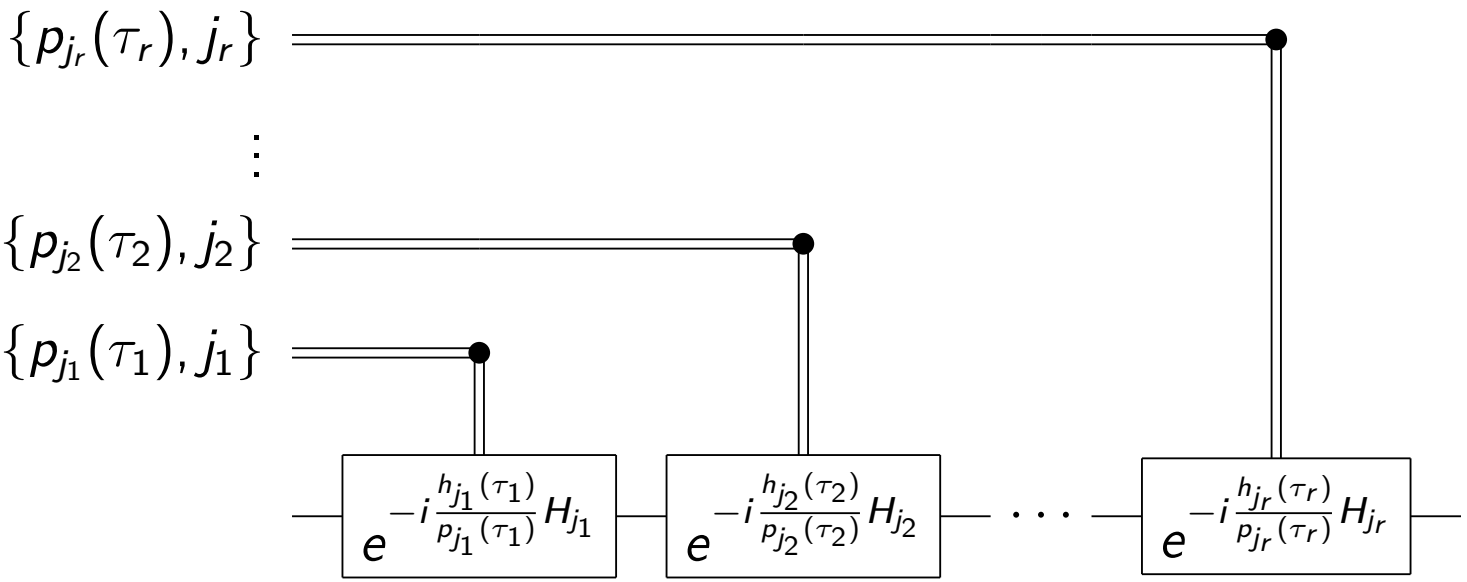
$$a = \left\| e^{-ih_j(\tau)H_j/p_j(\tau)} - \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}\tau \; H(\tau)\right)\right\| = O(t),$$

$$b = \left\| \int_0^t \mathrm{d}\tau \sum_j p_j(\tau) e^{-ih_j(\tau)H_j/p_j(\tau)} - \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}\tau \; H(\tau)\right)\right\| = O(t^2),$$

we see that the protocol is first-order accurate.

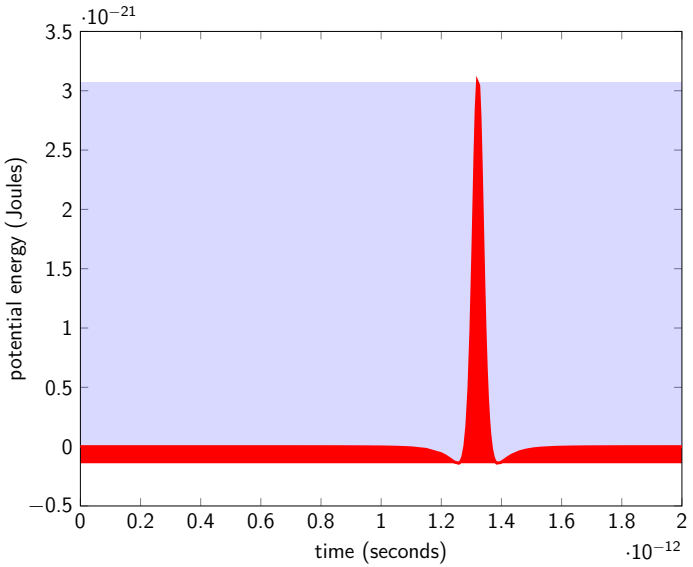
Complexity analysis

- A more careful calculation shows that $\{p_j(\tau), e^{-ih_j(\tau)H_j/p_j(\tau)}\}$ approximates $\{1, \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}\tau\ H(\tau)\right)\}$ with diamond-norm error $4(\int_0^t \mathrm{d}\tau\ \|h(\tau)\|_1)^2$.
- We can thus **use randomness to simulate time-dependent Hamiltonians with L^1-norm scaling.**
- To simulate for time t, divide the evolution into r steps and use the continuous quantum drift protocol within each step.



Applications to scattering theory

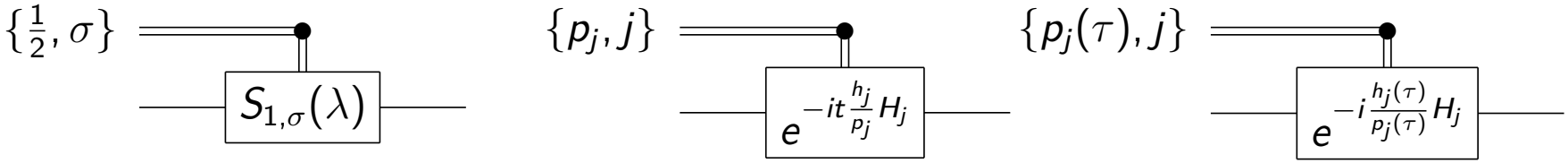
- The L^1-norm scaling result can be used in the semiclassical simulation of scattering processes in quantum chemistry.
- Target system: $H(t) = \sum_{j=1}^{n} h_j(t) H_j$, where H_j are both unitary and Hermitian.
- $\|h(\tau)\|_1 = O(N^{5/3})$ when the nuclear charges are dilute and $\|h(\tau)\|_1 = O(N^2)$ when they are concentrated. So the L^1-norm scaled algorithms can perform significantly better.



Outline

Summary

- Randomized quantum simulation algorithms:



- Randomization can be used
  - to improve the accuracy of product formulas;
  - to avoid the dependence on the number of terms in the Hamiltonian; and
  - to simulate time-dependent Hamiltonians with L^1-norm scaling.

Outlook

- Other applications of randomization: quantum simulation[9], optimization[10], ...
- Derandomizing randomized quantum simulation: Hamiltonian truncation[11],...
- Better analysis of randomized quantum algorithms: going beyond the triangle-inequality bound,...

[9] [Ouyang, White, Campbell 19]
[10] [Sweke et al. 19]
[11] [Tran et al. 19]