Exploiting anticommutation in Hamiltonian simulation

Qi Zhao and Xiao Yuan

1 Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, Maryland 20742, USA
2 Center on Frontiers of Computing Studies, Department of Computer Science, Peking University, Beijing 100871, China
3 Stanford Institute for Theoretical Physics, Stanford University, Stanford California 94305, USA

Quantum computing can efficiently simulate Hamiltonian dynamics of many-body quantum physics, a task that is generally intractable with classical computers. The hardness lies at the ubiquitous anticommutative relations of quantum operators, in corresponding with the notorious negative sign problem in classical simulation. Intuitively, Hamiltonians with more commutative terms are easier to simulate on a quantum computer, and anti-commutative relations generally cause more errors, such as in the product formula method. Here, we theoretically explore the role of anticommutative relation in Hamiltonian simulation. We find that, contrary to our intuition, anticommutative relations could also reduce the hardness of Hamiltonian simulation. Specifically, Hamiltonians with mutually anticommutative terms are easy to simulate, as what happens with ones consisting of mutually commutative terms. Such a property is further utilized to reduce the algorithmic error or the gate complexity in the truncated Taylor series quantum algorithm for general problems. Moreover, we propose two modified linear combinations of unitaries methods tailored for Hamiltonians with different degrees of anti-commutation. We numerically verify that the proposed methods exploiting anticommutative relations could significantly improve the simulation accuracy of electronic Hamiltonians. Our work sheds light on the roles of commutative and anti-commutative relations in simulating quantum systems.

Qi Zhao: zhaoq@umd.edu
Xiao Yuan: xiaoyuan@pku.edu.cn

1 Introduction

It is a notoriously hard problem to classically simulate an arbitrary large many-body physics systems. As proposed by Feynman in 1982 [20], quantum computers can directly encode a quantum system and hence efficiently simulate any quantum Hamiltonian dynamics. With the help of quantum computers, various quantum algorithms have been to proposed [5, 6, 8, 25, 26] and applied for studying different problems such as spin models [30], fermionic lattice models [36], quantum chemistry [2, 3, 35], and quantum field theories [23]. For a given Hamiltonian $H = \sum_{l=1}^{L} \alpha_l H_l$ with coefficients $\alpha_l$ and Pauli operators $H_l$, we focus on the simulation of the time evolution operator $U_0(t) = e^{-iHt}$, which is crucial for studying both the dynamic and static properties [1, 24, 27]. We consider the product formula methods [32] and ones based on linear-combination-of-unitaries (LCU) [16]. They have shown different asymptotical gate complexities with respect to the simulation time $t$, the accuracy $\varepsilon$, and the property of $H$.

For the product formula method [13–15, 17, 18, 33], also known as the Trotter-Suzuki algorithm, it divides the evolution time $t$ into small time steps $\delta t$ and approximates each $e^{-iH_l \delta t}$ via a product of $e^{-i\alpha_l H_l \delta t}$, for example as

$$e^{-iH\delta t} \approx \prod_l e^{-i\alpha_l H_l \delta t} + O(\delta t^2).$$

The error term $O(\delta t^2)$ is a pessimistic worst-case estimation, and it could be dramatically improved by considering commutative relations between different $H_l$. For example, when all $H_l$ are mutually commutative, the approximation becomes accurate and the error vanishes. We refer to Refs. [14, 17, 19, 21, 31, 34] for error analyses of product formula methods for more general problems. We note that due to the simplicity of the product formula method, it is widely
used in experimental implementations of Hamiltonian simulation with near-term quantum devices [4, 11, 12].

It is intuitively straightforward to see that the commutative relation could simplify or ease the complexity of Hamiltonian simulation whereas the anti-commutative relation could make the problem hard. While this seems to be true for the product formula method where Hamiltonians consisting of mutually (anti-)commutative terms have zero (maximal) approximation errors, the intuition breaks down in the LCU based methods. Specifically, we consider the truncated Taylor series algorithm [8], which, when expanding the operator $e^{-iH\delta t}$ to the Kth order, gives

$$e^{-iH\delta t} \approx \sum_{j=0}^{K} \frac{(\sum_l -i\alpha_l H_l \delta t)^j}{j!} + \mathcal{O}(\delta t^{K+1}).$$

The first summation term is realized via LCU and the approximation error comes from the truncated Taylor expansion error. It is not hard to see that the commutative relation could not universally reduce the approximation error as it does in the product formula method. Instead, the anti-commutative relation may cause cancellation of the higher order approximation error and hence be exploited to increase the simulation accuracy.

In this work, we investigate the role of anti-commutative relation in Hamiltonian simulation. We show that anti-commutative relation could also simplify the complexity of Hamiltonian simulation. We first use the anti-commutative relation to reduce the the truncated Taylor error in the Taylor series method, which lead to a tighter error analysis in contrast to the worse case one. Then we propose two modified LCU algorithms that exploit the anti-commutative relation and further reduce high-order truncation errors. Moreover, we show that Hamiltonians with mutually anti-commutative terms are perfectly simulable. We numerically test the performances of the proposed methods for electronic Hamiltonians and find significant improvements of the simulation accuracy with shallower quantum circuits. Our work thus broadens our understanding of commutative and anti-commutative relations in Hamiltonian simulation.

2 Preliminary

In this section, we review the product formula method and the Taylor series method. We show how the commutative relation plays an important role in the product formula method. In this work, we assume that the Hamiltonian $H$ is expressed as

$$H = \sum_{l=1}^{L} \alpha_l H_l,$$

where $H_l$ is a tensor product of Pauli operators and $\alpha_l$ are real coefficients. Each pair of elements $H_i$ and $H_j$ are either commutative or anti-commutative,

$$[H_i, H_j] = H_i H_j - H_j H_i = 0 \quad \text{(commutative)},$$

$$\{H_i, H_j\} = H_i H_j + H_j H_i = 0 \quad \text{(anti-commutative)}.$$

The target is to simulate the time evolution operator $U_0(t) = e^{-iHt}$ with quantum circuits.

2.1 Product formula method

The product formula (PF) approach approximates $U_0(t)$ via the product of exponentials of the individual operators $H_l$. By splitting the evolution time $t$ into $r$ segments ($r$ also known as the number of Trotter steps) and making $r$ sufficiently large, we can ensure the simulation is arbitrarily precise. In the first order PF method, we approximate $U_0(t)$ via a product form,

$$U_0(t) = e^{-i\sum_l \alpha_l H_l t} \approx U_1(t)$$

$$= \left( \prod_{l=1}^{L} e^{-i\alpha_l H_l t/r} \right)^r.$$

Recent error analyses [17, 19] showed that the approximation error is related to the commutative relation of the operators

$$\|U_1(t) - U_0(t)\| \leq \frac{t^2}{2r} \sum_{l_1=1}^{L} \left\| \sum_{l_2 \geq l_1}^{L} [\alpha_{l_2} H_{l_2}, \alpha_{l_1} H_{l_1}] \right\|.$$

where $\| \cdot \|$ corresponds to the spectral norm. The approximation error becomes smaller when more pairs of $H_{l_1}$ and $H_{l_2}$ commute, and the error vanishes when every term commutes to each other. Note that any two Pauli operators either commute or anti-commute, and thus the approximation error mainly originates from the anti-commutative relation.
2.2 Taylor series method

Now we review the truncated Taylor series (TS) method [8], which is a recently introduced more advanced Hamiltonian simulation algorithm that achieves different asymptotic error scaling. The main idea is to apply the (truncated) Taylor series of the evolution operator and implement the procedure for LCU. We rewrite the evolution operation in the Taylor series,

$$U_0(t) = e^{-iHt} = \sum_{k=0}^{\infty} \frac{(-iH)^k}{k!},$$

and the truncated Taylor series to the $K$-th order is

$$\tilde{U}(t) = \sum_{k=0}^{K} \frac{(-iH)^k}{k!},$$

which is a linear combination of unitaries $V_j = (-i)^kH_{1} \cdots H_{l_k}$ with the coefficients $\beta_j = \frac{i}{k!}\alpha_1 \cdots \alpha_{l_k}$. Without loss of generality, here we set each coefficient $\alpha_l > 0$ and thus each $\beta_j$ is positive.

We also need to divide evolution time $t$ into $r$ segments. In each segment, the algorithmic order of approximation mainly comes from the finite truncated Taylor series

$$\|\tilde{U}(t/r) - U_0(t/r)\| \leq e^{\alpha t/r} \left(\frac{\alpha t/r}{(K+1)!}\right)^K,$$

where $\alpha$ denotes the summation of $\alpha_l$, $\alpha = \sum_l \alpha_l$. Given a fixed total allowed error $\epsilon$, the truncated order parameter $K$ can be chosen as

$$K = O\left(\frac{\log \frac{\alpha t}{\epsilon}}{\log \frac{\alpha t}{\epsilon}}\right).$$

We refer to Appendix A and Ref.[8, 17] for details of the implementation and error analysis. From the above analysis, the commutative relation does not help much in reducing the truncation error. The truncation error exists even if the Hamiltonian consists of mutually commutative terms (as long as $H \neq 0$). In the next section, we show that it is actually the anti-commutative relation that plays a more important role in determining the error of the Taylor series method. We further show that the anti-commutative relation can be applied to reduce the algorithmic error and design modified Taylor series methods.

3 Anti-commutative cancellation

We first show the general idea of anti-commutative cancellation in refining the error analysis. We expand the time evolution operator $U_0(t) = e^{-iHt}$ as

$$U_0(t) = \sum_{m=0}^{\infty} \frac{(-it)^m}{m!} H^m,$$

where

$$H^m = \left(\sum_i \alpha_i H_i\right)^m = \sum_{l_1, \ldots, l_m=1}^{L} \alpha_{l_1} \alpha_{l_2} \cdots \alpha_{l_m} H_{l_1} H_{l_2} \cdots H_{l_m}.$$

Each $H^m$ has $L^m$ terms and its spectral norm is bound by

$$\|H^m\| \leq \left(\sum_{l=1}^{L} \alpha_l\right)^m = \alpha^m.$$

In the original error analysis, this upper bound $\alpha^m$ is used to estimate the truncation error, which is overestimated without considering the relations between different $H_l$. To refine the estimation of the norm $\|H^m\|$, we consider each sequence in Eq. (9)

$$H_{l_1} H_{l_2} \cdots H_{l_m} = H_{l_1} \cdots H_{l_m},$$

where $l_1, \ldots, l_m \in \{1, 2, \ldots, L\}$. If two terms, for instance $H_{l_1}$ and $H_{l_2}$ are anti-commutative with $\{H_{l_1}, H_{l_2}\} = 0$, the product with $H_{l_1} H_{l_2}$ at some positions can cancel out the terms with $H_{l_1} H_{l_1}$ at the same positions, for example as

$$\alpha_{l_1} \alpha_{l_2} \cdots \alpha_{l_m} (H_{l_1} H_{l_2} \cdots H_{l_m} + H_{l_1} H_{l_1} \cdots H_{l_m}) = 0.$$

After a given procedure of cancellation, part of $L^m$ terms in $H^m$ can be cancelled out via this anti-commutative relation and we rewrite $H^m$ as the sum of remaining sequences,

$$H^m = \sum_{r=1}^{R} \alpha_{l_{i_1}} \alpha_{l_{i_2}} \cdots \alpha_{l_{i_m}} H_{l_{i_1}} H_{l_{i_2}} \cdots H_{l_{i_m}}.$$
where \( l_1, \ldots, l_m \in \{1, \ldots, L\} \) and \( R \) is the number of remaining terms. For different methods of cancellation, we could have different \( R \). For example, if we use the cancellation pair by pair as shown in the following, for a pair-wisely anti-commutative Hamiltonian, \( R = L \). In order to estimate the norm of \( H^m \), we mainly focus on the sum of the coefficients in the remaining \( R \) terms. Thus we define a key parameter \( \alpha^{(m)} \) as

\[
\alpha^{(m)} = \min \sum_{r=1}^{R} \alpha_{l_1}^r \alpha_{l_2}^r \ldots \alpha_{l_m}^r,
\]

\[\text{s.t. } H^m = \sum_{r=1}^{R} \alpha_{l_1}^r \alpha_{l_2}^r \ldots \alpha_{l_m}^r H_{l_1} H_{l_2} \ldots H_{l_m},\]

with an upper bound of \( \| H^m \| \) as \( \| H^m \| \leq \alpha^{(m)} \).

Here the minimization is over all the possible cancellations.

In practice, it is generally hard to directly estimate \( \alpha^{(m)} \) for a general Hamiltonian. Nevertheless, we can repetitively apply a low order cancellation to serve as an upper bound for higher order cancellations,

\[
\alpha^{(km)} \leq (\alpha^{(k)})^m.\]

Here we take a second order cancellation as an example to show the cancellation procedure. We first focus on the positions \((1, 2)\) in a sequence of \( H_l \), and cancel out all the anti-commutative pairs on positions \((1, 2)\),

\[
H_1 H_2 \ldots H_m + H_2 H_1 \ldots H_m = 0, \quad \{H_1, H_2\} = 0.
\]

Then all the rest sequences satisfy

\[
H_1 H_2 \ldots H_m, \quad (l_1, l_2) \in \text{Comm},
\]

where \( \text{Comm} = \{(i, j)|[H_i, H_j] = 0\} \) denotes the set of commutative pairs and \( |\text{Comm}| \) denotes the number of elements in this set. We then repetitively cancel out the anti-commutative pairs on the positions \((3, 4), \ldots, (2P - 1, 2P)\) one by one, where \( P = \lfloor \frac{m}{2} \rfloor \). At last, the remaining terms satisfy that

\[
\begin{align*}
\text{Even } m & : H_1 H_{i_1} \ldots H_{i_{2P-1}} H_{i_{2P}}, \\
\text{Odd } m & : H_1 H_{i_1} \ldots H_{i_{2P-1}} H_{i_{2P}} H_{i_{2P+1}},
\end{align*}
\]

\[(i_{2j-1}, i_{2j}) \in \text{Comm}, \quad j = 1, \ldots, P.\]

Consequently, there are \( |\text{Comm}| \lfloor \frac{m}{2} \rfloor \), \( |\text{Comm}| \lfloor \frac{m}{2} \rfloor L \) terms left in \( H^m \) for even \( m \) and odd \( m \), and \( H^m \) can be expressed as

\[
H^m = \sum_{(l_2j-1, l_2j) \in \text{Comm}} \alpha_{l_1} \ldots \alpha_{l_m} H_{l_1} \ldots H_{l_m},
\]

Based on the above cancellation, we obtain the upper bounds for \( \alpha^{(m)} \)

\[
\begin{align*}
\alpha^{(2)} & \leq \alpha_{\text{comm}}, \\
\alpha^{(m)} & \leq |\frac{m}{2}| \alpha_{\text{comm}}, \quad \text{even } m, \\
\alpha^{(m)} & \leq \alpha_{\text{comm}}, \quad \text{odd } m,
\end{align*}
\]

where \( \alpha_{\text{comm}} = \sum_{(i,j) \in \text{comm}} \alpha_i \alpha_j \), and \( \alpha = \sum_{l=1}^{L} \alpha_l \) denote the summation of the coefficients of commutative pairs and the summation of coefficients of all \( L \) terms, respectively. As a result, the spectral norm of \( H^m \) is bounded as

\[
\begin{align*}
\| H^m \| & \leq \frac{m}{2} \alpha_{\text{comm}}, \quad \text{even } m, \\
\| H^m \| & \leq \alpha_{\text{comm}}, \quad \text{odd } m.
\end{align*}
\]

Similarly, we can directly estimate the third order \( \alpha^{(3)} \) and the fourth order cancellation \( \alpha^{(4)} \) numerically and use them to bound \( \alpha^{(m)} \).

4. Applications of anti-commutative cancellation

In this section, we show various applications of anti-commutative cancellation. We first use it to reduce the algorithmic error in the truncated Taylor series method in Sec. 4.1. Secondly, we propose a modified LCU algorithm to further reduce the high order error in the truncated Taylor expansion. Finally, we explore an extreme case where all the terms are pair-wisely anti-commutative and propose a modified LCU method tailored for this Hamiltonian. Note that the first two applications are suitable for all Hamiltonians, though different Hamiltonians would have different degree of improvement over the original method.

4.1 Tightening errors in truncated Taylor series algorithm

In this section, we show how the above anti-commutative cancellation could be applied to have a tighter estimation of the algorithmic error in the Taylor series method. In the original Taylor series method with expansion up to the
order terms may cancel out each other according to Eq. (22).

Here in the last line, we rewrite the terms \(-iHt\) and \(\sum_{j=2}^{\infty} \frac{(-itH)^j}{(K + j - 1)!/K!}\) as \(\sum_{l=1}^{L_i} \alpha_l(-itH_l) + \sum_{\tilde{e} \in H_{\tilde{e}}} \), respectively. Here \(H_{\tilde{e}}\) is the product of \(H_l\) that remains after a certain canceling procedure which can be expressed as

\[
H_{\tilde{e}} = (-i)^k H_{e_1} H_{e_2} \ldots H_{e_k}.
\]

In the original truncated Taylor LCU method, all these \(H_{\tilde{e}}\) are discarded as error terms. However, some \(H_{\tilde{e}}\) may equal to simple unitaries, such as identity \(I\), or unitaries that appears in lower order expansions, such as \(H_l\). For example, according to the anti-commutative cancellation, the second and third order terms \(H^2\) and \(H^3\) can be expressed as

\[
H^2 = \beta_0 I + \sum_{(l_1, l_2) \in \text{Comm}^*} \alpha_{l_1} \alpha_{l_2} H_{l_1} H_{l_2},
\]

\[
H^3 = \sum_{l} \beta_l H_l + \sum_{(l_1, l_2), (l_1, l_3), (l_2, l_3) \in \text{Comm}^*} \alpha_{l_1} \alpha_{l_2} \alpha_{l_3} H_{l_1} H_{l_2} H_{l_3},
\]

where we define the set containing all the commutative pairs without the same terms as \(\text{Comm}^* = \{(i, j)[H_i, H_j] = 0, i \neq j\}\). Here we use the facts \(H^2 = I\) and \(H_l H_{l'} H_{l'} = H_l\).

For high order terms, the following sequences equal to \(I\) and \(H_l\),

\[
\text{Even length : } (H_{l_1} H_{l_1}) \ldots (H_{l_p} H_{l_p}) = I,
\]

\[
\text{Odd length : } (H_{l_1} H_{l_1}) \ldots (H_{l_p} H_{l_p}) H_l = H_l.
\]
Instead of discarding all these terms, we can update the coefficients of $-iH_l$ from $a_l t$ to a new one $\gamma_l$ that includes higher order contributions and add the term $U_0 = -I$. Moreover, we can add other unitaries with the modified coefficients apart from the unitaries $-iH_l$ and $U_0 = -I$. For example, we choose $E$ number of unitaries with the largest coefficients in the rest of $H_E$; denoted as $U_1$ to $U_E$. Consequently, we construct an approximate $K$-th order term as

$$\tilde{R}_K = \frac{(-iH)^{K-1}}{K!} \sum_{l=1}^{L} \gamma_i(-iH_l) + \sum_{j=0}^{E} \gamma_j U_j ,$$

and we can implement

$$\tilde{U}(t) = \sum_{k=0}^{K-1} \frac{(-itH)^k}{k!} + \tilde{R}_K$$

via the LCU routine. Specifically, we implement the low order (less than $K - 1$) terms as usual and the $K$-th term $\tilde{R}_K$ with a modified circuit that involves little additional resource. The details of additional gates are shown in the numerical section. Overall, the refined method could reduce the error with only a few additional gates. It works well for Hamiltonians with large anti-commutative terms or very biased coefficients.

Now we analyze the error for the modified scheme. We divide the error into three parts $(K + 1)$-th order error $E_{K+1}$, $(K + 2)$-th order error $E_{K+2}$, and high order error $E_\infty$ as

$$\| \tilde{U}(t) - \sum_{k=0}^{\infty} \frac{(-itH)^k}{k!} \| = \| E_{K+1} + E_{K+2} + E_\infty \| .$$

(30)

Without loss of generality, we assume $K$ is odd. Benefitting from anti-commutative relation, we could obtain a more accurate analysis of $E_{K+1}$ and $E_{K+2}$ which take up a large proportion of the total error and are bound as

$$\| E_{K+1} \| \leq \frac{t^{K+1} \alpha_{comm} E_r}{(K + 1)!},$$

$$\| E_{K+2} \| \leq \frac{t^{K+2} \alpha_{comm} \alpha_r^{(3)}}{(K + 2)!},$$

(31)

where $E_r$ is the sum of coefficients of the remaining terms in $\sum_{(l_1,l_2) \in Comm} \alpha_{l_1} \alpha_{l_2} H_{l_1} H_{l_2}$ in Eq. (26), which are not included in $U_j$ and $-iH_l$ and $E_r \leq \alpha_{comm} - (\sum \alpha_l^2)$, and $\alpha_r^{(3)}$ is the sum of coefficients $\alpha_{l_1}, \alpha_{l_2}, \alpha_{l_3}$ in $H^3$ according to Eq. (26).

We could directly use the bound in Eq. (23) for high order terms $\| E_\infty \|$. Though we could also calculate the sequences which equals to $H_l$ or $I$ in $E_\infty$, it would not influence the results a lot.

With the parameter $\frac{\alpha}{\sqrt{\alpha_{comm}}} = q$, the error can be expressed as

$$\| \tilde{U}(t) - \sum_{k=0}^{\infty} \frac{(-itH)^k}{k!} \| \leq \frac{(t\alpha)^{K+1} E_r}{(K + 1)! q^{K} \alpha^2} + \frac{(t\alpha)^{K+2} \alpha_r^{(3)}}{(K + 2)! q^{K-1} \alpha^3} + \frac{(t\alpha)^{K+3} (q + 1)e^{\alpha/q} + (\alpha - 1)(q - 1)e^{\alpha q}}{(K + 3)!}.$$  

(32)

We refer to Appendix A and C for details of the implementation and error analysis. From the above expression, we know that this error is related to some key parameters $\alpha_{comm}$, $E_r$ and $\alpha_r^{(3)}$.

Though this modified method could not influence asymptotic performance, it could significantly reduce the error in the near term implementation of the Taylor series method. The numerical verification of the improvement on performance is shown in the next section.

4.3 Modified Taylor series method with pair-wisely anti-commutative terms

At last, we consider an extreme case with Hamiltonians consisting of pair-wisely anti-commutative terms. For example, we construct an $n$-qubit Hamiltonian $H = \sum_{j=1}^{n} H_j$ with $n$ pair-wisely anti-commutative terms,

$$H_1 = X_1, H_2 = Z_1 \otimes Z_2, H_j = Z_1 \otimes X_2 \cdots \otimes X_{j-1} \otimes Z_j, j = 3, \ldots , n.$$  

(33)

In this section, we show how the evolution of this kind of Hamiltonians can be simulated via the LCU method without any approximation error. We first show the exact form of the expansion of the Hamiltonian.

**Lemma 1.** For a Hamiltonian $H = \sum_{l=1}^{L} \alpha_l H_l$ where all the terms $H_l$ are unitary and mutually anti-commutative, $\{H_{l_1}, H_{l_2}\} = 0$, $l_1 \neq l_2$, $H^m$ can be expressed as

$$H^m = \gamma_0^{(m)} I + \sum_{l=1}^{L} \gamma_l^{(m)} H_l,$$

(34)

for some coefficients $\gamma_l^{(m)}$, $l = 0, \ldots, L$. 

Accepted in Quantum 2021-08-24, click title to verify. Published under CC-BY 4.0.
In particular, we have
\[
\gamma_0^{(2j-1)} = 0, \quad \gamma_0^{(2j)} = \beta_s^{2j-2} \alpha_l, \\
\gamma_0^{(2j+1)} = \beta_s^{2j}, \quad \gamma_0^{(2j+2)} = 0,
\]
where \( j \in \mathbb{N}^+ \) is a positive integer and \( \beta_s = \sqrt{\sum_{l=1}^L \alpha_l^2} \). Now, we expand the evolution \( U(t) = e^{-iHt} \), and we can rewrite it as
\[
U(t) = \tilde{U}(t) = \tilde{\alpha}_0 I + \sum_l \tilde{\alpha}_l (-iH_l),
\]
where
\[
\tilde{\alpha}_0 = 1 - \frac{\beta_s^2 t^2}{2!} + \frac{\beta_s^4 t^4}{4!} + \cdots = \cos(t\beta_s), \quad \tilde{\alpha}_l = \frac{\alpha_l}{\beta_s} \sin(t\beta_s), \quad l = 1, \ldots, L.
\]
Note that here the coefficients \( \tilde{\alpha}_l \) may not be always positive, and we could replace them with the absolute values and change the sign of the unitary gate accordingly. The sum of the absolute value coefficients is
\[
s = \sum_l |\tilde{\alpha}_l| = |\cos(t\beta_s)| + \frac{\alpha_l}{\beta_s} |\sin(t\beta_s)|,
\]
which is a function of \( t \) and \( \beta_s \). In particular, we have \( L\beta_s^2 \geq \alpha^2 \) and thus \( 1 \leq s \leq 1 + \sqrt{L} \).

**Theorem 1.** For a Hamiltonian \( H = \sum_{l=1}^L \alpha_l H_l \) acting on \( n \) qubits where all the terms \( H_l \) are mutually anti-commutative, \( \{H_{l_1}, H_{l_2}\} = 0, \ l_1 \neq l_2 \), we can implement the evolution \( U(t) = e^{-iHt} \) without algorithmic error via the LCU method with the gate complexity \( \mathcal{O}(L^2 (n + \log L)) \).

Here the gate complexity \( \mathcal{O}(L^2 (n + \log L)) \) is independent of \( t \) and \( \epsilon \). The details of the implementation of LCU is shown in Appendix D.2. Moreover, the expansion of the unitary also indicates efficient classical simulation of the Hamiltonian similar to Hamitlonians with mutually commutative terms. We can also extend the above strict case to more general cases.

**Corollary 1.** Suppose there exists a positive integer \( M \), satisfying that \( H^M = \gamma I \), we can implement the evolution \( U(t) = e^{-iHt} \) without algorithmic error via the LCU method.

Similarly, we rewrite the unitary as
\[
U(t) = \sum_{k=0}^\infty \frac{(-itH)^k}{k!} = \sum_{k=0}^{M-1} \sum_{j=0}^{\infty} \frac{(-it)^j M \gamma_j}{(k+jM)!} \gamma_k (-itH)^k,
\]
where coefficients \( \gamma_k = \sum_{j=0}^\infty \frac{(-it)^j M \gamma_j}{(k+jM)!} \). We can implement this unitary like the original truncated Taylor method with at most \( M - 1 \) order without any algorithmic error. In the above analysis, we require that the Hamiltonians are mutually anti-commutative, which is rather strict.

**Corollary 2.** When the Hamiltonian \( H \) is not perfectly anti-commutative, but close to the perfect case,
\[
H^2 = \beta_s^2 I + V, \quad \|V\| \leq \varepsilon_A.
\]
There exist coefficients \( \tilde{\alpha}_l \), such that \( \tilde{U}(t) \) is close to \( U_0(t) = \sum_{k=0}^\infty \frac{(-itH)^k}{k!} \) with an error \( \mathcal{O}(t^2 \varepsilon_A) + \mathcal{O}(t^3 \varepsilon_A \alpha) \) for a small \( t \),
\[
\|\tilde{U}(t) - \sum_{k=0}^\infty \frac{(-itH)^k}{k!}\| = \mathcal{O}(t^2 \varepsilon_A) + \mathcal{O}(t^3 \varepsilon_A \alpha).
\]
Note that here we need a small time \( t \). Thus for a long time \( t \), we could also divide it into \( r \) segments, and the total error is \( \mathcal{O}(t^2 \varepsilon_A/r) + \mathcal{O}(t^3 \varepsilon_A \alpha/r^2) \). When \( t = n \), and \( \alpha = \mathcal{O}(n) \), then the error is \( \mathcal{O}(n^2 \varepsilon_A/r) \). The parameter for commutative pairs has the upper bound, \( \varepsilon_A \leq \sum_{(i,j)\in\text{Comm}} \alpha_i \alpha_j < \alpha_{\text{comm}} \). Compared to the error in the first order PF \( \mathcal{O}(n^2 \alpha_{\text{anti}}/r) \), the differences are the prefactors quantifying the anti-commutation and commutation property. When the Hamiltonian is more anti-commutative, \( \alpha_{\text{comm}} < \alpha_{\text{anti}} \) then this approximate LCU method could has the advantage over the first order PF.

### 5 Numerical simulation

Here in this section, we show the numerical performance of the proposed methods. Lattice Hamiltonians mostly consist of low weight terms, which are likely to be commutative to each
other; whereas chemistry Hamiltonians, after the fermion-to-qubit encoding, have more anti-commutative terms. Since the proposed methods are more suitable for an anti-commutative Hamiltonian, we focus on the electronic structure of chemistry Hamiltonians. The second quantized form of the electronic Hamiltonians is

\[ H = \sum_{p,q} h_{pq} \alpha_p^\dagger \alpha_q + \frac{1}{2} \sum_{p,q,r,s} h_{pqrs} \alpha_p^\dagger \alpha_r^\dagger \alpha_s \alpha_q, \quad (42) \]

where \( \alpha_p^\dagger \) and \( \alpha_p \) are creation and annihilation operators [27]. Then it is converted to a qubit Hamiltonian via the Jordan-Wigner transformation [22].

\[
\alpha_p = \frac{X_p + iY_p}{2} \otimes Z_{p-1} \otimes \cdots \otimes Z_0, \\
\alpha_p^\dagger = \frac{X_p - iY_p}{2} \otimes Z_{p-1} \otimes \cdots \otimes Z_0.
\]

We consider 23 different molecules as shown in Table 1 and the Hamiltonian data is from Ref. [28].

5.1 Refined errors estimation

In the Taylor series method, the total time is divided into several segments with each time segment \( t = \ln 2/\alpha \). The error in the original analysis is

\[ \varepsilon_\alpha = \frac{\delta_2^2 + 3\delta_0 + 4}{2} \delta_0, \quad \delta_0 = 2^{(\ln 2)^2 K + 1}/(K + 1)! . \quad (44) \]

where \( \delta_0 \) is the truncated error in the Taylor expansion. According to Eq. (23), we calculate the refined errors \( \varepsilon_n \) and show the improvement of the refined error estimation for various molecules with up to 25 qubits. In order to show the improvement clearly, we calculate the ratios \( \varepsilon_\alpha/\varepsilon_n \) versus different truncated order \( K = 10, 20, 30, 40 \) in Table 1 using the second order cancellation. Moreover, we also plot the ratio \( \varepsilon_\alpha/\varepsilon_n \) for some example molecules, HO, LiH, BH, and BeH\(_2\) using second order cancellation, third order, and fourth order cancellation in Fig. 1 using the upper bounds in Eq. (61). The lower bounds of the key parameters \( q = \alpha/(\alpha^{(2)})^{1/2}, \alpha/(\alpha^{(3)})^{1/3}, \alpha/(\alpha^{(4)})^{1/4} \) can be estimated numerically, which are summarized in Table 2. Though higher order cancellation brings more significant improvement, the numerical estimation of high order cancellation also becomes much harder with the increase of the number of terms \( L \) and the number of qubits \( n \).

<table>
<thead>
<tr>
<th>Formula</th>
<th>Qubits</th>
<th>( K = 10 )</th>
<th>( K = 20 )</th>
<th>( K = 30 )</th>
<th>( K = 40 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>HO</td>
<td>11</td>
<td>1.445</td>
<td>2.016</td>
<td>2.813</td>
<td>3.926</td>
</tr>
<tr>
<td>LiH</td>
<td>11</td>
<td>1.866</td>
<td>3.285</td>
<td>5.782</td>
<td>10.177</td>
</tr>
<tr>
<td>BH</td>
<td>11</td>
<td>1.962</td>
<td>3.615</td>
<td>6.660</td>
<td>12.270</td>
</tr>
<tr>
<td>BeH(_2)</td>
<td>13</td>
<td>1.990</td>
<td>3.714</td>
<td>6.930</td>
<td>12.933</td>
</tr>
<tr>
<td>NH(_2)</td>
<td>13</td>
<td>1.655</td>
<td>2.611</td>
<td>4.119</td>
<td>6.500</td>
</tr>
<tr>
<td>BH(_2)</td>
<td>13</td>
<td>1.062</td>
<td>1.411</td>
<td>4.817</td>
<td>16.123</td>
</tr>
<tr>
<td>CH(_3)</td>
<td>15</td>
<td>1.919</td>
<td>3.466</td>
<td>6.260</td>
<td>11.305</td>
</tr>
<tr>
<td>CH(_4)</td>
<td>17</td>
<td>2.492</td>
<td>5.708</td>
<td>13.074</td>
<td>29.948</td>
</tr>
<tr>
<td>NO</td>
<td>19</td>
<td>1.428</td>
<td>1.972</td>
<td>2.734</td>
<td>3.761</td>
</tr>
<tr>
<td>BN</td>
<td>19</td>
<td>1.561</td>
<td>2.337</td>
<td>3.498</td>
<td>5.237</td>
</tr>
<tr>
<td>LiOH</td>
<td>21</td>
<td>1.639</td>
<td>2.565</td>
<td>4.013</td>
<td>6.627</td>
</tr>
<tr>
<td>HBO</td>
<td>21</td>
<td>1.663</td>
<td>2.637</td>
<td>4.180</td>
<td>6.627</td>
</tr>
<tr>
<td>HO</td>
<td>21</td>
<td>1.525</td>
<td>2.324</td>
<td>3.273</td>
<td>4.796</td>
</tr>
<tr>
<td>CHF</td>
<td>21</td>
<td>1.616</td>
<td>2.497</td>
<td>3.857</td>
<td>5.958</td>
</tr>
<tr>
<td>H(_2)NO</td>
<td>23</td>
<td>1.532</td>
<td>2.254</td>
<td>3.317</td>
<td>4.881</td>
</tr>
<tr>
<td>CH(_2)O</td>
<td>23</td>
<td>1.582</td>
<td>2.396</td>
<td>3.630</td>
<td>5.499</td>
</tr>
<tr>
<td>NO(_2)F</td>
<td>23</td>
<td>1.574</td>
<td>2.374</td>
<td>3.581</td>
<td>5.401</td>
</tr>
<tr>
<td>CH(_2)F</td>
<td>23</td>
<td>1.605</td>
<td>2.465</td>
<td>3.785</td>
<td>5.811</td>
</tr>
<tr>
<td>CH(_3)F</td>
<td>25</td>
<td>1.575</td>
<td>2.375</td>
<td>3.583</td>
<td>5.405</td>
</tr>
<tr>
<td>CH(_3)Li</td>
<td>25</td>
<td>2.014</td>
<td>3.799</td>
<td>7.166</td>
<td>13.518</td>
</tr>
<tr>
<td>OCH(_3)</td>
<td>25</td>
<td>1.705</td>
<td>2.763</td>
<td>4.480</td>
<td>7.263</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Molecule</th>
<th>( \alpha/(\alpha^{(2)})^{1/2} )</th>
<th>( \alpha/(\alpha^{(3)})^{1/3} )</th>
<th>( \alpha/(\alpha^{(4)})^{1/4} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>HO</td>
<td>1.0339</td>
<td>1.0552</td>
<td>1.0699</td>
</tr>
<tr>
<td>LiH</td>
<td>1.0582</td>
<td>1.0949</td>
<td>1.1177</td>
</tr>
<tr>
<td>BH</td>
<td>1.0630</td>
<td>1.0994</td>
<td>1.1222</td>
</tr>
<tr>
<td>BeH(_2)</td>
<td>1.0644</td>
<td>1.1061</td>
<td>1.1324</td>
</tr>
</tbody>
</table>

Table 2: Parameters \( \alpha/(\alpha^{(2)})^{1/2}, \alpha/(\alpha^{(3)})^{1/3}, \alpha/(\alpha^{(4)})^{1/4} \) estimation for second order, third order, and fourth order cancellation.

5.2 Reducing the high order error in modified algorithm

In Section 4.2, we further reduce the high order truncated error via modifying the coefficients and involving more unitary operations in the Taylor series. We calculate three different types of new errors with \( t = \ln 2/\alpha \):

1. Refined errors shown in Eq. (23) using second order cancellation;
2. Reduced errors in modified method without additional unitaries \( (E = 0) \) in Eq. (32).
3. Reduced errors in modified method with \( E = 2^w - L - 1 \) additional unitaries in Eq. (32) where \( w = \lceil \log_2 L \rceil \).

Table 1: Error Reduction using second-order cancellation \( \varepsilon_\alpha/\varepsilon_n \) for various molecules.
Figure 1: Ratio $\varepsilon_0/\varepsilon_n$ with second order, third order, and fourth order cancellation versus different truncated order $K$ with $t = \ln \frac{2}{\alpha}$ for molecules HO, LiH, BH, BeH$_2$.

According to Appendix C and Ref. [17], these three schemes we applied in the numerical simulation have almost the same gate cost. In order to show the improvement clearly, we show the curves of the ratio of the original error to new calculated error, $\varepsilon_0/\varepsilon_n$, versus different truncated order $K$ for molecules, BH, CH$_4$, HBO, CH$_3$Li with terms $L$ ranging from 631 to 19548. We find that with increasing $K$, the reduced error becomes more significant. Note that here the improvement is not related to the system size $n$ nor the number of terms $L$, but related to the anti-commutative property of different molecules.

In some Hamiltonian simulation tasks, a predetermined simulation accuracy $\varepsilon$ is given. The task is to find the most efficient circuit approximating the ideal evolution with at most $\varepsilon$ error. Thus, we also compare the required truncated $K$ with a given simulation accuracy $\varepsilon$ for molecules BH, CH$_4$, HBO, CH$_3$Li. We choose the simulation time $t = n$ where $n$ is the number of qubits. We find that in most of cases shown in Table 3, the refined error scheme or the modified scheme could reduce one order of Taylor expansion. Although this does not seem a significant change in theory, it can indeed reduce both the gate count and the number of ancillary qubits, making its practical implementation much easier. For a Hamiltonian with $L$ unitaries, one order reduction could reduce $\log L$ number of ancillary qubits. We show the total CNOT gates and ancillary qubits reduction for these examples in Table 4. For example, consider a large molecule Hamiltonian with $N = 100$ spin orbitals and $h = 50$ electrons, we have about $L \approx N^2 h^2 = 2.5 \times 10^7$ terms and in one segment our method can roughly reduce 25 qubits and accordingly remove $7.5 \times 10^8$ CNOT gates that act on those qubits.

6 Discussion

In this work, we explore the anti-commutative relation in Hamiltonian simulation. We find that Anti-commutative relation plays a positive role in the truncated Taylor series method. We show that the simulation accuracy could be significantly improved by utilizing the anti-commutative relation. In particular, pair-wisely anti-commutative Hamiltonians admit an efficient and error free simulation with the LCU method. Our work is the first attempt of exploiting anti-commutative relations in Hamiltonian reduction, which may inspire and motivate more interesting and significant future works to find more efficient reduction mechanisms that combine with other techniques or considering specific Hamiltonians that have large anti-commutative terms.

When we consider some specific types of Hamiltonians, the performance of Taylor series method could be further improved. For instance, combining our result with techniques in Ref. [29], we may further reduce the simulation error for electronic structure Hamiltonians. Here most of our results are based on a low order anti-
Table 3: Minimum $K$ for required accuracy in three different schemes for molecules BH, CH$_4$, HBO, CH$_3$Li. The first column for each molecule is with original Taylor series method. The second column is with refined error scheme in Sec. 4.1. And the third column is with modified scheme in Sec. 4.2 with $E = 2^n - L - 1$.

<table>
<thead>
<tr>
<th>Error $\varepsilon$</th>
<th>BH</th>
<th>CH$_4$</th>
<th>HBO</th>
<th>CH$_3$Li</th>
</tr>
</thead>
<tbody>
<tr>
<td>1e-06</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>1e-07</td>
<td>11</td>
<td>10</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>1e-08</td>
<td>12</td>
<td>11</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>1e-09</td>
<td>13</td>
<td>12</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>1e-10</td>
<td>14</td>
<td>13</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>1e-11</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>1e-12</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>1e-13</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>1e-14</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>1e-15</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>1e-16</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>1e-17</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>1e-18</td>
<td>19</td>
<td>19</td>
<td>20</td>
<td>19</td>
</tr>
<tr>
<td>1e-19</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>1e-20</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 4: Number of CNOT gates and ancillary qubits reduction for one order reduction of Taylor expansion for simulating molecules BH, CH$_4$, HBO, CH$_3$Li with simulation time $t = n$ where $n$ is the number of qubits.

<table>
<thead>
<tr>
<th>Reduction</th>
<th>BH</th>
<th>CH$_4$</th>
<th>HBO</th>
<th>CH$_3$Li</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNOT Gate</td>
<td>$8.7 \times 10^6$</td>
<td>$3.0 \times 10^6$</td>
<td>$1.5 \times 10^6$</td>
<td>$2.3 \times 10^6$</td>
</tr>
<tr>
<td>Ancillary qubits</td>
<td>10</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>

which involves the summation of the products of different sequences [13, 14, 18]. Meanwhile, due to the complementary advantages of commutation in product formula and anti-commutation in the Taylor series method, we could also consider hybrid approaches of Trotter and Taylor series methods. In this case, the Hamiltonian could be divided into commutative and anti-commutative terms (sets), which are simulated using Trotter and Taylor expansion respectively. Of course, how to divide the terms and whether such a hybrid approach is useful seem a very interesting future direction.

The error reduction and speed-up in this work mainly come from representing the power of Hamiltonian $H$ with the summation of fewer unitary terms. Thus, the cancellation idea is not restricted to anti-commutative cancellation. For instance, if the Hamiltonian $H$ has certain symmetry properties, then any order of $H$ could be expressed in the summation of the terms in the symmetry subspace, which might be useful for designing more efficient Hamiltonian simulation algorithms.

Note that the perfect anti-commutation case admits an efficient classical simulation of Hamiltonian evolution. And the Trotter formula, for instance the second order trotter method is used in the classical algorithms of estimating the partition function [9, 10]. We expect the anti-commutation property may inspire better classical simulation method of quantum problems.

7 Acknowledgement

We are grateful to Andrew Childs, Di Fang, Lin Lin, Yuan Su, and You Zhou for useful discussions and anonymous referees for their comments on an earlier draft. We thank Richard Meister for providing the molecular Hamiltonian data. QZ acknowledges the support by the Department of Defense through the Hartree Postdoctoral Fellowship at QuICS. XY acknowledges support from the Simons Foundation.

A Taylor expansion method

In order to implement $\tilde{U}(t)$, we need three key elements: a state preparation procedure accomplished by oracle $G$, a reflection denoted as $R$, and an oracle selecting unitary operations denoted as select($V$).
We assume that the ancillary system is a $d$ dimensional Hilbert space $\mathcal{H}_{an}$ and the evolution operation $e^{-iHt}$ acts on an $n$ qubit Hilbert space $\mathcal{H}_s$ with dimension $2^n$.

Here the oracle $G$ is a unitary that transforms the ancillary state $|0\rangle$ to

$$G |0\rangle = \frac{1}{\sqrt{s}} \sum_j \sqrt{\beta_j} |j\rangle,$$  \hspace{1cm} (45)

where $s$ is the normalized parameter,

$$s = \sum_j \beta_j = \sum_{k=0}^{K} \frac{(t \sum_l a_l)^k}{k!}.$$  \hspace{1cm} (46)

The select($V$) is defined as

$$\text{select}(V) = \sum_j |j\rangle \langle j| \otimes V_j.$$  \hspace{1cm} (47)

The control register $|j\rangle \langle j|$ acts on the ancillary system $\mathcal{H}_{an}$. We construct an operator as

$$W = (G^\dagger \otimes I) \text{select}(V)(G \otimes I),$$  \hspace{1cm} (48)

which when applied to the input state $|0\rangle \otimes |\Psi\rangle$ gives

$$W(|0\rangle \otimes |\Psi\rangle) = \frac{1}{s}(|0\rangle \otimes \tilde{U}(t) |\Psi\rangle) + \sqrt{\frac{s^2-1}{s}} |\phi\rangle,$$  \hspace{1cm} (49)

where $|\phi\rangle$ is orthogonal to the subspace span$\{ |0\rangle \otimes \mathcal{H}_s \}$. After projecting to $|0\rangle$ of the ancillary system $\mathcal{H}_{an}$, we can obtain the approximate evolution $\tilde{U}(t) |\Psi\rangle$ with probability $1/s$. However, the failure probability is nonignorable, thus we need to boost the coefficient before $|0\rangle \otimes U(t) |\Psi\rangle$. When $s = 2$, we could use the oblivious amplitude amplification to boost the probability,

$$-WRW^\dagger RW(|0\rangle \otimes |\Psi\rangle) = |0\rangle \otimes \tilde{U}(t) |\Psi\rangle,$$  \hspace{1cm} (50)

where $R$ is the reflection $R = (I - 2 |0\rangle \langle 0| \otimes I$ acting on $\mathcal{H}_{an} \otimes \mathcal{H}_s$. Here oblivious amplitude amplification is similar to the iterations in the Grover algorithm. The operation $RW$ can be understood as the reflection along the axis vector $|0\rangle \otimes U(t) |\Psi\rangle$. After this, we implement $-WRW^\dagger = 2W |0\rangle \langle 0| W^\dagger - I$, which is the reverse reflection along the axis $W(|0\rangle \otimes |\Psi\rangle)$ and completes the amplification. Thus the post selected evolution $\tilde{V}(t)$ is

$$\tilde{V}(t) = (|0\rangle \otimes I) - WRW^\dagger RW(|0\rangle \otimes I).$$  \hspace{1cm} (51)

In general, $s$ can be far away from 2. Similar to the product formula method, here we could divide the simulation time $t$ into $r$ segments such that the simulation time in each segment $\tau = t/r = \frac{\ln 2}{\sum_l a_l}$ satisfies

$$s = \sum_{k=0}^{K} \frac{(t \sum_l a_l/r)^k}{k!} \approx \exp \left( t \sum_l a_l/r \right) = 2.$$  \hspace{1cm} (52)

For the remaining time $\tau_{re}$ less than $\tau$, $\tau = \frac{\ln 2}{\sum_l a_l}$, $t = (r - 1)\tau + \tau_{re}$. Another ancillary qubit can be used to exactly boost the value of $s$ to 2, see Refs [8, 17]. Thus the whole evolution can be implemented by

$$\tilde{U}_{TS}(t) = \tilde{V}(\tau_{re})\tilde{V}(\tau)^{r-1}.$$  \hspace{1cm} (53)

The algorithmic error mainly comes from the finite truncated Taylor series

$$||\tilde{U}(\tau) - U_0(\tau)|| \leq 2 \frac{(\ln 2)^{K+1}}{(K+1)!}.$$  \hspace{1cm} (54)
It can be verify that
\[ \tilde{V}(\tau) = \frac{3}{8} \tilde{U}(\tau) - \frac{4}{8^3} \tilde{U}(\tau) \tilde{U}(\tau) \tilde{U}(\tau). \] (55)

Supposing the truncated error \( \delta = 2^{(\ln 2)^{K+1}/(K+1)} \), the distance for each segment can be bound as
\[ \| \tilde{V}(\tau) - U_0(\tau) \| \leq \frac{\delta^2 + 3\delta + 4}{2}, \] (56)
and the total error accumulates with \( r \) segments as
\[ \| \tilde{V}(\tau_{re}) \tilde{V}(\tau)^{r-1} - e^{-iHt} \| \leq \frac{\delta^2 + 3\delta + 4}{2} \delta^r. \] (57)

Given a fixed total allowed error \( \varepsilon \), the error for each segment is \( O(\varepsilon/r) \). In order to make Eq. (56) satisfy the error constraint, the parameter \( K \) can be chosen as \( K = O\left(\frac{\log \frac{\alpha t}{\varepsilon}}{\log \log \frac{\alpha t}{\varepsilon}}\right) \), where \( \alpha = \sum t \alpha_t \). The segment number is \( r = \frac{\alpha t}{\ln 2} \). In each segment, we need to implement \( G \) and \( \text{select}(V) \). The oracle \( G \) can be implemented by a unary encoding which needs \( O(KL) \) gates. And \( \text{select}(V) \) mainly contributes to the gate complexity, which can be implemented via \( K \) controlled-select\((H) \) requiring the gate cost \( O(KL(n + \log L)) \). Thus the total gate complexity is \( O\left(\alpha tL(n + \log L)\frac{\log \frac{\alpha t}{\varepsilon}}{\log \log \frac{\alpha t}{\varepsilon}}\right) \). We refer to Refs [8, 17] for more details of this algorithm.

B Reducing the errors in the original LCU scheme

Here we give the refined error estimation for the Taylor series method using the second order, third order cancelling method. The fourth order cancelling is similar. In the second order cancellation, we can divide the high order terms into two parts: odd order terms and even order terms. And the spectral norm of each part can be bound via Eq. (21),

\[
\begin{align*}
\left\| \tilde{U}(t) - \sum_{k=0}^{\infty} \frac{(-itH)^k}{k!} \right\| & \leq \sum_{j=0}^{\infty} \left| \sum_{k=K+1, \, k \equiv j \mod{2}}^{\infty} \frac{(-itH)^k}{k!} \right| \\
& \leq \sum_{j=0, 1} \left( \sum_{k=K+1, \, k \equiv j \mod{2}}^{\infty} \frac{t^k \sqrt{\alpha_{\text{comm}}}}{k!} \alpha^j \right) \\
& \leq \left( \frac{t \sqrt{\alpha_{\text{comm}}}}{(K + 1)!} \right)^{K+1} \sum_{j=0, 1} \sum_{l=0, \, K+1+l \equiv j \mod{2}} \frac{q^j t^l \sqrt{\alpha_{\text{comm}}}}{(K + 1)!} \\
& = \left( \frac{t \sqrt{\alpha_{\text{comm}}}}{(K + 1)!} \right)^{K+1} \left[ q \left( e^{t \sqrt{\alpha_{\text{comm}}}} + (-1)^K e^{-t \sqrt{\alpha_{\text{comm}}}} \right) + \left( e^{t \sqrt{\alpha_{\text{comm}}}} + (-1)^K e^{-t \sqrt{\alpha_{\text{comm}}}} \right) \right] \\
& = \frac{(ta)^{K+1}}{(K + 1)!} \left( q + 1 \right) e^{t \sqrt{\alpha_{\text{comm}}}} + \left( (-1)^K (q - 1) e^{-t \sqrt{\alpha_{\text{comm}}}} \right) \\
& \leq \frac{(ta)^{K+1}}{(K + 1)!} \left( q + 1 \right) e^{t \sqrt{\alpha_{\text{comm}}}} + \left( (-1)^K (q - 1) e^{-t \sqrt{\alpha_{\text{comm}}}} \right) \\
& \leq \frac{(ta)^{K+1}}{(K + 1)!} \left( q + 1 \right) e^{t \sqrt{\alpha_{\text{comm}}}} + \left( (-1)^K (q - 1) e^{-t \sqrt{\alpha_{\text{comm}}}} \right).
\end{align*}
\] (58)

Here we give the explicit form of the error via third order cancellation. For simplicity, we suppose \( K = 2 \mod{3} \) and denote \( \frac{\alpha}{t^{(3)^j} \sqrt{3}} = q_3 \) and the trivial parameters \( q_0 = q_1 = 1 \). Then the error are
divided into three parts as follows. The cases $K = 0$ or $1 \pmod{3}$ are similar.

$$\left\| \tilde{U}(t) - \sum_{k=0}^{\infty} \frac{(-iH)^k}{k!} \right\| \leq \sum_{j=0,1,2} \left\| \sum_{k=K+1, k=j \pmod{3}}^{\infty} \frac{(-iH)^k}{k!} \right\|$$

$$\leq \sum_{j=0,1,2} \sum_{k=K+1, k=j \pmod{3}}^{\infty} \frac{t^k (\alpha^{(j)})^{\frac{k}{3}}}{k!} \alpha^{(j)}$$

$$= \frac{(t\alpha)^{K+1} S_0 + q_3 S_1 + \frac{q_5}{q_2^2} S_2}{(K+1)!},$$

where

$$S_0 = \sum_{l=0 \pmod{3}, l \geq 0} \frac{t^l}{l!} (\alpha^{(3)})^{l/3} = \frac{1}{3} \left( e^{\alpha_{l/3} / q_3} + e^{\omega \alpha_{l/3} / q_3} + e^{\omega^2 \alpha_{l/3} / q_3} \right),$$

$$S_1 = \sum_{l=1 \pmod{3}, l \geq 0} \frac{t^l}{l!} (\alpha^{(3)})^{l/3} = \frac{1}{3} \left( e^{\alpha_{l/3} / q_3} + \omega^2 e^{\omega \alpha_{l/3} / q_3} + \omega e^{\omega^2 \alpha_{l/3} / q_3} \right),$$

$$S_2 = \sum_{l=2 \pmod{3}, l \geq 0} \frac{t^l}{l!} (\alpha^{(3)})^{l/3} = \frac{1}{3} \left( e^{\alpha_{l/3} / q_3} + \omega e^{\omega \alpha_{l/3} / q_3} + \omega^2 e^{\omega^2 \alpha_{l/3} / q_3} \right).$$

For a $p$-th order cancellation, we denote the parameter $\alpha^{(p)}_{l/p} = q_p$, and could give a general result,

$$\left\| \tilde{U}(t) - \sum_{k=0}^{p-1} \frac{(-iH)^k}{k!} \right\| \leq \sum_{j=0}^{p-1} \left\| \sum_{k=K+1, k=j \pmod{p}}^{\infty} \frac{(-iH)^k}{k!} \right\|$$

$$\leq \sum_{j=0}^{p-1} \sum_{k=K+1, k=j \pmod{p}}^{\infty} \frac{t^k (\alpha^{(j)})^{\frac{k}{p}}}{k!} \alpha^{(j)}$$

$$= \frac{(t\alpha)^{K+1} e^{\alpha_{l/3} / q_p}}{(K+1)!} \max_j \left( \frac{q_p}{q_j} \right)^j \sum_{l \geq 0} \frac{t^l}{l!} (\alpha^{(j)})^{l/p}$$

$$\leq \frac{(t\alpha)^{K+1} e^{\alpha_{l/3} / q_p}}{(K+1)!} \max_j \left( \frac{q_p}{q_j} \right)^j.$$

C Details of modified LCU Algorithm reducing high order error

In the modified LCU method, we construct an approximate $K$-th order term as

$$\tilde{R}_K = \frac{(-iH)^{K-1}}{K!} \left[ \sum_{l=1}^{L} \gamma_l (-iH_l) + \sum_{j=0}^{E} \gamma_j U_j \right].$$
According to the anti-commutative cancellation, the second and third order terms $H^2$ and $H^3$ in $\sum_{e} \tilde{\gamma}_e H_{\tilde{e}}$ are

\[
H^2 = \beta_0 I + \sum_{(l_1, l_2) \in \text{Comm}^*} \alpha_{l_1} \alpha_{l_2} H_{l_1} H_{l_2},
\]

\[
H^3 = \sum_{l} \beta_l H_l + \sum_{(l_1, l_2), (l_1, l_3), (l_2, l_3) \in \text{Comm}^*} \alpha_{l_1} \alpha_{l_2} \alpha_{l_3} H_{l_1} H_{l_2} H_{l_3}
\]

(63)

where $\text{Comm}^* = \{(i, j) | [H_i, H_j] = 0, i \neq j\}$. Here the first line $H^2$ is directly from the anti-commutative cancellation. For $H^3$ terms, when $l_1, l_2, l_3$ are equal $l_1 = l_2 = l_3$ or two of them are equal, for example, $l_1 = l_2 \neq l_3$. Then we have $H_{l_1} H_{l_2} H_{l_3} = H_{l_3}$. When $l_1, l_2, l_3$ are different, there are two cases: 1. All the pairs are commutative, 2. There is at least one pair $H_{l_1}$ and $H_{l_2}$ which are anti-commutative. For case 2, all the possible six sequences can be cancelled pair by pair as

\[
\begin{align*}
H_{l_1} H_{l_2} H_{l_3} &+ H_{l_2} H_{l_1} H_{l_3} = 0, \\
H_{l_1} H_{l_3} H_{l_2} &+ H_{l_3} H_{l_1} H_{l_2} = 0, \\
H_{l_2} H_{l_1} H_{l_3} &+ H_{l_1} H_{l_2} H_{l_3} = 0.
\end{align*}
\]

(64)

Then only the terms $H_{l_1} H_{l_2} H_{l_3}$ with $(l_1, l_2), (l_1, l_3), (l_2, l_3) \in \text{Comm}^*$ (case 1) are remained.

The upper bound of $(K + 1)$-th order terms is straightforward according to the definition of $E_\varepsilon$, which are the terms not included in $U_j$ and $-i H_l$. For the $K + 2$-th order terms, $\beta_l H_l$ are already included into $\gamma_l (-iH_l)$ and the remaining terms are

\[
E_{K+2} = \frac{t^{K+2}}{(K + 2)!} (\langle -i H \rangle)^{K-1} \sum_{(l_1, l_2), (l_1, l_3), (l_2, l_3) \in \text{Comm}^*} \alpha_{l_1} \alpha_{l_2} \alpha_{l_3} (-i)^3 H_{l_1} H_{l_2} H_{l_3}.
\]

(65)

We denote the coefficient

\[
\alpha^{(3)}_r = \sum_{(l_1, l_2), (l_1, l_3), (l_2, l_3) \in \text{Comm}^*} \alpha_{l_1} \alpha_{l_2} \alpha_{l_3} i H_{l_1} H_{l_2} H_{l_3},
\]

(66)

and we have

\[
\| E_{K+2} \| \leq \frac{t^{K+2}}{(K + 2)!} \| H^{K-1} \| \| \alpha^{(3)}_r \|.
\]

(67)

When $K - 1$ is even, we use the bound

\[
\| H^{K-1} \| \leq \frac{K-1}{\alpha_{\text{comm}}}
\]

(68)

in Eq. (21). We could also use the similar method to carefully calculate the rest high order part $\| E_{\infty} \|$, but it would not influence the error a lot. Thus we directly use the bounds in Eq. (21) and obtain

\[
\| E_{\infty} \| \leq \frac{t^{K+3}}{(K + 3)!} \sqrt{\alpha_{\text{comm}}} \left[ \frac{\alpha}{\sqrt{\alpha_{\text{comm}}}} \left( \sum_{l \text{ odd}, l \geq 0} \frac{t^l}{l!} \sqrt{\alpha_{\text{comm}}} \right) + \sum_{l \text{ even}, l \geq 0} \frac{t^l}{l!} \sqrt{\alpha_{\text{comm}}} \right].
\]

(69)

In the modified scheme, all the low order $K - 1$ order terms can be implemented as usual. To implement modified $\hat{R}_K$, we need to replace the $K$-th controlled select$(H)$ with one controlled select$(H')$, where select$(H')$ has $E + 1 + L$ unitaries to select, which is more complicate than select$(H)$,

\[
\begin{align*}
\text{select}(H) &= \sum_{l=1}^{L} \langle l \rangle \langle l \rangle \otimes (-i H_l), \\
\text{select}(H') &= \sum_{l=1}^{L} \langle l \rangle \langle l \rangle \otimes (-i H_l) + \langle L + 1 \rangle \langle L + 1 \rangle \otimes (-I) \\
&+ \sum_{j=1}^{E} \langle L + 1 + j \rangle \langle L + 1 + j \rangle \otimes U_j.
\end{align*}
\]

(70)
In Ref. [17], a select($H$) with $L$ different unitaries to select requires $7.5 \cdot 2^w + 6w - 26$ CNOT gates and $7.5 \cdot 2^w + 6w - 28$ T gates, where $w = \lceil \log_2 L \rceil$. Then we can see that when $E \leq 2^w - L - 1$, select($H$) and select($H'$) have the same gate cost. Thus, in the numerical simulation, we implement the modified schemes with $E = 0, 2^w - L - 1$, which indicates almost no additional gate cost.

D Modified LCU method for the pair-wisely anti-commutative Hamiltonians

D.1 Proof of Lemma 1

We can prove it iteratively. The conclusion is true for $m = 1$. Now, suppose it is true for $m > 1$, we have

$H^{m+1} = H^m H$

$$= \left( \gamma_0^{(m)} I + \sum_{l=1}^{L} \gamma_l^{(m)} H_l \right) \left( \sum_{l=1}^{L} \alpha_l H_l \right)$$

$$= \sum_{l=1}^{L} \gamma_l^{(m)} \alpha_l I + \gamma_0^{(m)} \sum_{l=1}^{L} \alpha_l H_l + \sum_{l_1 \neq l_2} \gamma_l^{(m)} \alpha_{l_1} \alpha_{l_2} H_{l_1} H_{l_2}$$

(71)

The third term is 0 due to anti-commutative relation and we have

$$\gamma_0^{(m+1)} = \sum_l \gamma_l^{(m)} \alpha_l, \gamma_l^{(m+1)} = \gamma_l^{(m)} \alpha_l.$$  \hspace{1cm} (72)

Thus $H^{m+1} = \gamma_0^{(m+1)} I + \sum_{l=1}^{L} \gamma_l^{(m)} H_l$.

D.2 Implementation of LCU

From Lemma 1, we can expand the evolution $U(t) = e^{-iHt}$ as

$$U(t) = \bar{U}(t) = \bar{\alpha}_0 I + \sum_l \bar{\alpha}_l (-iH_l),$$

(73)

where

$$\bar{\alpha}_0 = 1 - \beta_s^2 t^2 / 2! + \beta_s^4 t^4 / 4! + \ldots = \cos(t \beta_s),$$

$$\bar{\alpha}_l = \alpha_l t - \alpha_l \beta_s^2 t^3 / 3! + \alpha_l \beta_s^4 t^5 / 5! + \ldots = \frac{\alpha_l}{\beta_s} \sin(t \beta_s), \ l = 1, \ldots, L.$$  \hspace{1cm} (74)

$$\beta_s = \sqrt{\sum_{l=1}^{L} \alpha_l^2}.$$

Now we show how to implement the evolution via the LCU method. We show two different methods here. The first method is via direct implementation of $U(t)$ via LCU. Similar to the original LCU method, we need oracles $G$, select($U$),

$$G \ket{0} = \frac{1}{\sqrt{s}} \sum_j \sqrt{\bar{\alpha}_j} \ket{j} ,$$

$$\text{select}(U) = \sum_l \ket{l} \bra{l} \otimes U_l ,$$

(75)

$$W = (G^\dagger \otimes I) \text{select}(U) (G \otimes I) ,$$

$$s = \sum_l |\bar{\alpha}_l| = |\cos(t \beta_s)| + \frac{\alpha}{\beta_s} |\sin(t \beta_s)| .$$

Here we choose $U_0 = I$ or $-I$, $U_l = -iH_l$ or $iH_l$ according to the sign of $\bar{\alpha}_l$. According to Ref. [8], select($U$) can be implemented via select($U_l$) one by one (totally $L + 1$). And each select($U_l$) requires
Thus we choose the first segment as $t_1 = \left\lfloor \frac{t}{\pi^2} \right\rfloor \beta_s$ such that $s = 1$. The rest evolution time is $t_2 = t - t_1 \leq \frac{\pi}{\beta_s}$ and the corresponding $s$ can still be very large. When $\frac{s^2}{\beta_s^2} \geq 3$, we choose another simulation time $t_{seg}$ as

$$t_{seg} = \frac{\arcsin \left( \frac{2 \alpha}{\beta_s} - \sqrt{\frac{\alpha^2}{\beta_s^2} - 3} \right)}{\beta_s}$$

such that

$$s = | \cos(t_{seg}\beta_s)| + \frac{\alpha}{\beta_s} |\sin(t_{seg}\beta_s)| = 2.$$  

(78)

And $t_2$ is divided into $r$ segments $r = \left\lfloor \frac{t_{seg}}{t_{rest}} \right\rfloor$, $t_2 = rt_{seg} + t_{rest}$, $t_{rest} \leq t_{seg}$. In the rest time, we have

$$1 < s = | \cos(t_{rest}\beta_s)| + \frac{\alpha}{\beta_s} |\sin(t_{rest}\beta_s)| < 2.$$  

(79)

Thus we need $r+2$ segments, where $r$ can be bound as

$$r \leq \frac{\pi}{\arcsin \left( \frac{2 \alpha}{\beta_s} - \sqrt{\frac{\alpha^2}{\beta_s^2} - 3} \right)} = \mathcal{O}(\sqrt{L}).$$

(80)

The last equation is due to $\frac{s^2}{\beta_s^2} \leq L$.

When $\frac{s^2}{\beta_s^2} < 3$, we also have $1 < s < 2$. For all the segments where $s < 2$, we can use another anicilla to convert $s < 2$ into the $s = 2$ case [8, 17]. And all the $s = 2$ segments can be implemented via the oblivious amplitude amplification as the usual LCU method. Consequently, the upper bound of gate cost is $r\mathcal{O}(L(n + \log L)) = \mathcal{O}(L^{\frac{3}{2}}(n + \log L))$. We could implement this digital simulation efficiently without algorithmic error via standard LCU method.

### D.3 Proof of Corollary 2

Here we consider the case where the Hamiltonian $H$ is close to the perfectly anti-commutative,

$$H^2 = \beta_s^2 I + V, \quad ||V|| \leq \varepsilon_A.$$  

(81)

Then the unitary evolution can be expressed as

$$U_0(t) = \sum_{k=0}^{\infty} \frac{(-itH)^k}{k!} = \sum_{\text{even } k \geq 0} \frac{t^k}{k!}(-\beta_s^2 I - V)^{\frac{k}{2}} + \sum_{\text{odd } k \geq 1} \frac{t^k}{k!}(-\beta_s^2 I - V)^{\frac{k-1}{2}} \sum_l (-iH_l).$$

(82)
Note that here $V$ is not unitary but the sum of a few different unitaries. Here we also implement the operator $\tilde{U}(t)$ in the perfect case in Eq. (36)

$$\tilde{U}(t) = \tilde{\alpha}_0 I + \sum_l \tilde{\alpha}_l (-iH_l),$$

and the error can be bound by

$$\left\| \tilde{U}(t) - \sum_{k=0}^{\infty} \frac{(-itH)^k}{k!} \right\| \leq \sum_{k \geq 2, \text{even}} \frac{t^k}{k!} \left( (\beta_s^2 + \varepsilon_A)^{\frac{k}{2}} - \beta_s^k \right) + \sum_{k \geq 3, \text{odd}} \frac{t^k}{k!} \alpha \left( (\beta_s^2 + \varepsilon_A)^{\frac{k}{2}} - \beta_s^{k-1} \right) \leq \frac{1}{2} \left[ e^{t\sqrt{\beta_s^2 + \varepsilon_A}} + e^{-t\sqrt{\beta_s^2 + \varepsilon_A}} - \left( e^{t\beta_s} + e^{-t\beta_s} \right) \right] + \frac{1}{2} \frac{\alpha}{\sqrt{\beta_s^2 + \varepsilon_A}} \left( e^{t\sqrt{\beta_s^2 + \varepsilon_A}} - e^{-t\sqrt{\beta_s^2 + \varepsilon_A}} - \frac{\alpha}{\beta_s} \left( e^{t\beta_s} - e^{-t\beta_s} \right) \right),$$

which is roughly $O(t^2 \varepsilon_A) + O(t^3 \varepsilon_A \alpha)$ with a small $t$. The first inequality is because high order of $\beta_s^2 I$ are all included in the coefficient $\gamma_0$.

References


