Nearly optimal lattice simulation by product formulas – Supplementary Material

Andrew M. Childs and Yuan Su
Department of Computer Science, Institute for Advanced Computer Studies, and Joint Center for Quantum Information and Computer Science, University of Maryland
(Dated: July 4, 2019)

I. NONRIGOROUS ERROR ANALYSIS BY BAKER-CAMPBELL-HAUSDORFF FORMULA

In this section, we review an approach to product-formula error analysis based on the Baker-Campbell-Hausdorff (BCH) formula [1]. We explain why this argument is difficult to formalize and how local error analysis overcomes the difficulty.

Let $H=\sum_{j=1}^{n-1}H_{j,j+1}$ be an n-qubit lattice Hamiltonian, so the ideal evolution under H for time t is given by e^{-itH}. We group terms in an even-odd pattern as $H_{\mathrm{odd}}:=H_{1,2}+H_{3,4}+\cdots=\sum_{k=1}^{\frac{n}{2}}H_{2k-1,2k}$, $H_{\mathrm{even}}:=H_{2,3}+H_{4,5}+\cdots=\sum_{k=1}^{\frac{n}{2}-1}H_{2k,2k+1}$. For simplicity, we only analyze the first-order product formula $e^{-itH_{\mathrm{even}}}e^{-itH_{\mathrm{odd}}}$, which approximates the ideal evolution with error

$$e^{-itH_{\mathrm{even}}}e^{-itH_{\mathrm{odd}}}-e^{-itH}. \tag{1}$$

Jordan, Lee, and Preskill analyzed the scaling of this product-formula error as follows [1]. They first apply the BCH formula to the product formula and rewrite

$$e^{-itH_{\mathrm{even}}}e^{-itH_{\mathrm{odd}}}=e^{-itH-\frac{t^2}{2}\left[H_{\mathrm{even}},H_{\mathrm{odd}}\right]+i\frac{t^3}{12}\left[H_{\mathrm{even}},\left[H_{\mathrm{even}},H_{\mathrm{odd}}\right]\right]-i\frac{t^3}{12}\left[H_{\mathrm{even}},\left[H_{\mathrm{odd}},H_{\mathrm{even}}\right]\right]+\cdots}. \tag{2}$$

Expanding the Taylor series and ignoring all higher-order terms, they obtain

$$e^{-itH_{\mathrm{even}}}e^{-itH_{\mathrm{odd}}}\approx e^{-itH}-\frac{t^2}{2}\left[H_{\mathrm{even}},H_{\mathrm{odd}}\right]. \tag{3}$$

They thus estimate

$$\left\|e^{-itH_{\mathrm{even}}}e^{-itH_{\mathrm{odd}}}-e^{-itH}\right\|\approx O\big(\left\|\left[H_{\mathrm{even}},H_{\mathrm{odd}}\right]\right\|t^2\big)=O\big(nt^2\big), \tag{4}$$

which is the desired error scaling for the first-order product formula [2, Eq.(3)].

To formalize this argument, we must also consider higher-order terms. For a pth-order term in the Taylor series, we would instead estimate the spectral norm of a nested commutator

$$\left\|\left[H_{\mathrm{odd}}\left[\cdots,\left[H_{\mathrm{even}},H_{\mathrm{odd}}\right]\cdots\right]\right]\right\|t^p. \tag{5}$$

By locality, this commutator scales like $O(nt^p)$ as long as the number of nesting layers is constant. However, when the number of layers is larger than n, the scaling becomes $O(n^pt^p)$. The n-dependence is now superlinear, which does not provide the desired error scaling [2, Eq.(3)]. See [3, Appendix B] for further discussion of this issue and drawbacks of this approach.

In comparison, local error analysis gives

$$\begin{aligned}&e^{-itH_{\mathrm{even}}}e^{-itH_{\mathrm{odd}}}-e^{-itH}\\=&\int_0^t\mathrm{d}\tau_1\int_0^{\tau_1}\mathrm{d}\tau_2\ e^{-i(t-\tau_1)H}e^{-i\tau_1H_{\mathrm{even}}}e^{i\tau_2H_{\mathrm{even}}}\left[-iH_{\mathrm{even}},\ -iH_{\mathrm{odd}}\right]e^{-i\tau_2H_{\mathrm{even}}}e^{-i\tau_1H_{\mathrm{odd}}}.\end{aligned} \tag{6}$$

By the triangle inequality, we have

$$\left\|e^{-itH_{\mathrm{even}}}e^{-itH_{\mathrm{odd}}}-e^{-itH}\right\|\leq\int_0^t\mathrm{d}\tau_1\int_0^{\tau_1}\mathrm{d}\tau_2\left\|\left[H_{\mathrm{even}},H_{\mathrm{odd}}\right]\right\|=O\big(\left\|\left[H_{\mathrm{even}},H_{\mathrm{odd}}\right]\right\|t^2\big)=O\big(nt^2\big). \tag{7}$$

Similar to the analysis based on the BCH formula, we are effectively bounding the lowest-order error, but the analysis is now done in a fully rigorous way. Generalizations give similar (though more complicated) error expressions for higher-order product-formulas, which we discuss in detail in Section II, Section III, and Section IV.

II. CANONICAL PRODUCT FORMULAS AND ORDER CONDITIONS

In this section, we introduce notation and terminology that is useful for studying higher-order formulas. Similar to the first-order case, it is instructive to study a setting where the Hamiltonian is the sum of two Hermitian terms $H = A + B$. The evolution of H for time t is given by the unitary operator $\mathscr{E}(t) = e^{-itH}$, which may then be simulated using a specific product-formula algorithm, such as the Lie-Trotter formula or the Suzuki formulas. We will not analyze these formulas case-by-case. Instead, we consider *canonical product formulas*, a universal concept that includes well-known constructions.

Definition 1 (Canonical product formula). *Let H be a Hamiltonian consisting of two terms $H = A + B$, where A and B are Hermitian operators. We say that an operator-valued function $\mathscr{S}(t)$ is a* canonical product formula *for $H = A + B$ if it has the form*

$$\mathscr{S}(t) := \mathscr{S}_s(t) \cdots \mathscr{S}_2(t)\mathscr{S}_1(t) = \big(e^{-itb_sB}e^{-ita_sA}\big) \cdots \big(e^{-itb_2B}e^{-ita_2A}\big)\big(e^{-itb_1B}e^{-ita_1A}\big), \tag{8}$$

where $a_1, \ldots, a_s$ and $b_1, \ldots, b_s$ are real coefficients. The parameter s denotes the number of stages, and $\mathscr{S}_j(t) = e^{-itb_jB}e^{-ita_jA}$ is the jth-stage operator for $j = 1, \ldots, s$. We let u be an upper bound on the coefficients, i.e.,

$$\max\{|a_1|, \ldots, |a_s|, |b_1|, \ldots, |b_s|\} \leq u. \tag{9}$$

Finally, we say that the product formula $\mathscr{S}(t)$ has order p for some integer $p \geq 1$ if

$$\mathscr{S}(t) = \mathscr{E}(t) + O(t^{p+1}). \tag{10}$$

We call $\mathscr{S}(t)$ an (s, p, u)-formula if we need an explicit description of the parameters.

Although common constructions of product formulas involve stages where exponentials can be ordered both as $e^{-itb_jB}e^{-ita_jA}$ and as $e^{-ita_jA}e^{-itb_jB}$, we can achieve such orderings by padding with identity operators. In particular, we now show in detail how some well-known constructions of product formulas can be recast in the canonical form.

Example 1 (First-order formula). *The first-order formula $e^{-itB}e^{-itA}$ may be represented as a 1-stage canonical formula by setting $b_1 = a_1 = 1$, whereas its reversed version $e^{-itA}e^{-itB}$ is a 2-stage canonical formula with the choice $b_2 = 0, a_2 = b_1 = 1, a_1 = 0$.*

Example 2 (Second-order formula). *The second-order formula $e^{-i\frac{t}{2}A}e^{-itB}e^{-i\frac{t}{2}A}$ may be represented as a 2-stage canonical formula by setting $b_2 = 0, a_2 = \frac{1}{2}, b_1 = 1, a_1 = \frac{1}{2}$, whereas its reversed version $e^{-i\frac{t}{2}B}e^{-itA}e^{-i\frac{t}{2}B}$ is a 2-stage canonical formula with the choice $b_2 = \frac{1}{2}, a_2 = 1, b_1 = \frac{1}{2}, a_1 = 0$.*

Example 3 ($(2k)$th-order formula). *The $(2k)$th-order Suzuki formula $\mathscr{S}_{2k}(t)$ defined in [2, Eq.(2)] is an (s, p, u)-formula, where $s \leq 2 \cdot 5^{k-1}$, $p = 2k$, and $u = 1$.*

We now study the order conditions of a product formula. (Similar order conditions are sketched in [4], but we discuss them here for completeness.) Whenever possible, we follow the notation and terminology of [5]. We need the following lemma.

Lemma 1. *Let $F(t)$ be an operator-valued function that is infinitely differentiable. Let $p \geq 1$ be a nonnegative integer. The following two conditions are equivalent.*

1. *Asymptotic scaling: $F(t) = O(t^{p+1})$.*
2. *Derivative condition: $F(0) = F'(0) = \cdots = F^{(p)}(0) = 0$.*

Proof. Condition 2 implies 1 by Taylor's theorem. Assuming Condition 1 holds, we must have that

$$\|F(t)\| \leq C_1 t^{p+1} \tag{11}$$

for some $C_1 \geq 0$ (and for t sufficiently small). Let $0 \leq j \leq p$ be the first integer such that $F^{(j)}(0) \neq 0$. We use Taylor's theorem to find $C_2 \geq 0$ such that

$$\|F(t)\| \geq \left\|F^{(j)}(0)\right\| \frac{t^j}{j!} - C_2 t^{j+1}. \tag{12}$$

We combine the above inequalities and divide both sides by t^j. Taking the limit $t \to 0$ gives us a contradiction. $\square$

By definition, a product formula $\mathscr{S}(t)$ has order p for some integer $p \geq 1$ if

$$\mathscr{S}(t) = \mathscr{E}(t) + O(t^{p+1}) \tag{13}$$

holds for any $H = A + B$. Invoking Lemma 1, we find an equivalent order condition

$$\mathscr{S}^{(j)}(0) = (-iH)^j \tag{14}$$

for $0 \leq j \leq p$.

As in the first-order case, we seek an integral representation of the product-formula error $\mathscr{S}(t) - \mathscr{E}(t)$. To this end, we differentiate $\mathscr{S}(t)$ and rewrite the derivative as $\frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}(t) = (-iH)\mathscr{S}(t) + \mathscr{R}(t)$, where

$$\mathscr{R}(t) := \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}(t) - (-iH)\mathscr{S}(t). \tag{15}$$

Recall that $\mathscr{S}(t)$ is accurate up to order $p \geq 1$. Therefore, $\mathscr{S}(0) = I$ and we obtain the initial value problem $\frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}(t) = (-iH)\mathscr{S}(t) + \mathscr{R}(t),\ \mathscr{S}(0) = I$. The solution of this problem is given by the variation-of-constants formula

$$\mathscr{S}(t) - \mathscr{E}(t) = \int_0^t e^{-i(t-\tau)H}\mathscr{R}(\tau)\mathrm{d}\tau. \tag{16}$$

We now determine an order condition for the operator $\mathscr{R}(t)$. Since $\mathscr{S}(t)$ is at least first-order accurate, we have $\mathscr{S}^{(1)}(0) = -iH$ and therefore $\mathscr{R}(0) = \mathscr{S}^{(1)}(0) - (-iH)\mathscr{S}(0) = 0$. By taking derivatives iteratively, one can show that

$$\mathscr{R}(0) = \mathscr{R}^{(1)}(0) = \cdots = \mathscr{R}^{(p-1)}(0) = 0. \tag{17}$$

Conversely, if higher-order derivatives of $\mathscr{R}$ satisfy the above condition, we must have

$$\mathscr{S}^{(j)}(0) = (-iH)\mathscr{S}^{(j-1)}(0) = \cdots = (-iH)^j\mathscr{S}(0) \tag{18}$$

for $1 \leq j \leq p$. Using the fact that $\mathscr{S}(0) = I$, we have $\mathscr{S}^{(j)}(0) = (-iH)^j$ for $0 \leq j \leq p$. Therefore, our new order condition (17) is equivalent to (14).

We proceed to rewrite the integrand $\mathscr{R}$ using the product formula $\mathscr{S}$ and another operator. Specifically, we let $\mathscr{T}(t)$ be the operator such that

$$\mathscr{R}(t) = \mathscr{S}(t)\mathscr{T}(t). \tag{19}$$

In quantum simulation, the product formula $\mathscr{S}(t)$ is unitary and therefore $\mathscr{T}(t) = \mathscr{S}(t)^\dagger\mathscr{R}(t)$. However, we will see that $\mathscr{T}(t)$ has significantly richer structure than it might seem. Analyzing the combinatorial structure of $\mathscr{T}(t)$ will be the central topic of the next section. For now, we shall focus on its order condition.

We claim that (17) is equivalent to the order condition

$$\mathscr{T}^{(j)}(0) = 0 \qquad \text{for all } 0 \leq j \leq p-1. \tag{20}$$

By the general Leibniz rule

$$\mathscr{R}^{(j)}(0) = (\mathscr{S}\mathscr{T})^{(j)}(0) = \sum_{l=0}^{j}\binom{j}{l}\mathscr{S}^{(j-l)}(0)\mathscr{T}^{(l)}(0), \tag{21}$$

so (20) implies (17). We prove the converse by induction. For $j = 0$, we have $\mathscr{R}(0) = 0$ and $\mathscr{S}(0) = I$. Therefore, $\mathscr{R}(0) = \mathscr{S}(0)\mathscr{T}(0)$ implies that $\mathscr{T}(0) = 0$. Assume that $\mathscr{T}^{(l)}(0) = 0$ has been proved for all $0 \leq l \leq j \leq p-2$. We apply the general Leibniz rule to compute the $(j+1)$th-order derivative of $\mathscr{R}$ and find

$$0 = \mathscr{R}^{(j+1)}(0) = \sum_{l=0}^{j+1}\binom{j+1}{l}\mathscr{S}^{(j+1-l)}(0)\mathscr{T}^{(l)}(0) = \mathscr{S}(0)\mathscr{T}^{(j+1)}(0) = \mathscr{T}^{(j+1)}(0). \tag{22}$$

Therefore $\mathscr{T}^{(l)}(0) = 0$ for all $0 \leq l \leq j+1$.

We now summarize all the product-formula order conditions determined above in the following theorem.

Theorem 1 (Order conditions for canonical product formulas)**.** *Let H be a Hamiltonian consisting of two terms $H = A + B$, where A and B are Hermitian operators. Let $p \geq 1$ be an integer and let $\mathscr{S}(t)$ be a canonical product formula for $H = A + B$. The following four conditions are equivalent.*

1. $\mathscr{S}(t) = e^{-itH} + O(t^{p+1})$.

2. $\mathscr{S}^{(j)}(0) = (-iH)^j$ *for all* $0 \leq j \leq p$.

3. *There is some infinitely differentiable operator-valued function* $\mathscr{R}(t)$ *with* $\mathscr{R}^{(j)}(0) = 0$ *for all* $0 \leq j \leq p - 1$*, such that*
$$\mathscr{S}(t) = e^{-itH} + \int_0^t e^{-i(t-\tau)H} \mathscr{R}(\tau) \mathrm{d}\tau. \tag{23}$$

4. *There is some infinitely differentiable operator-valued function* $\mathscr{T}(t)$ *with* $\mathscr{T}^{(j)}(0) = 0$ *for all* $0 \leq j \leq p - 1$*, such that*
$$\mathscr{S}(t) = e^{-itH} + \int_0^t e^{-i(t-\tau)H} \mathscr{S}(\tau)\mathscr{T}(\tau) \mathrm{d}\tau. \tag{24}$$

Furthermore, the operator-valued functions $\mathscr{R}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}(t) - (-iH)\mathscr{S}(t)$ *and* $\mathscr{T}(t) = \mathscr{S}(t)^\dagger \mathscr{R}(t)$ *are uniquely determined.*

Proof. We have already proved $1 \Leftrightarrow 2$, $2 \Rightarrow 3$, and $3 \Rightarrow 4$, except for the differentiability of $\mathscr{R}$ and $\mathscr{T}$, which follows trivially from the definitions $\mathscr{R}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}(t) - (-iH)\mathscr{S}(t)$ and $\mathscr{T}(t) = \mathscr{S}(t)^\dagger \mathscr{R}(t)$.

Assume Condition 3 holds for some $\mathscr{R}(t)$. Differentiation gives
$$\mathscr{S}'(t) = (-iH)e^{-itH} + (-iH)e^{-itH} \int_0^t e^{i\tau H} \mathscr{R}(\tau)\mathrm{d}\tau + e^{-itH}e^{itH}\mathscr{R}(t) = (-iH)\mathscr{S}(t) + \mathscr{R}(t). \tag{25}$$

Therefore, $\mathscr{R}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}(t) - (-iH)\mathscr{S}(t)$ is uniquely determined, and $3 \Rightarrow 2$ follows from our previous analysis. In a similar way, we can show that $\mathscr{T}(t) = \mathscr{S}(t)^\dagger \big(\frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}(t) - (-iH)\mathscr{S}(t)\big)$ is uniquely determined and $4 \Rightarrow 3$ thus follows. $\square$

III. SIMPLIFIED LOCAL ERROR REPRESENTATION

In Section II, we found an integral representation for the product-formula error $\mathscr{S}(t) - \mathscr{E}(t) = \int_0^t e^{-i(t-\tau)H}\mathscr{R}(\tau)\mathrm{d}\tau$. A direct Taylor expansion of $\mathscr{R}(t)$ gives the correct scaling in t but an incorrect dependence on n. To address this issue, we introduced an auxiliary operator $\mathscr{T}(t)$.

A direct Taylor expansion of $\mathscr{T}(t)$ based on its definition $\mathscr{T}(t) = \mathscr{S}(t)^\dagger \mathscr{R}(t)$ does not give the correct n-dependence either. Instead, we construct an alternative expression for the integrand that consists of a linear combination of nested commutators, where the number of commutators and the number of nested layers are both independent of n and t. Such an expression is referred to as a *local error representation* in [5]. To this end, we compute $\mathscr{R}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}(t) - (-iH)\mathscr{S}(t)$ explicitly. We then perform unitary conjugation to create $\mathscr{S}(t)$ on the left-hand side of $\mathscr{R}(t)$. Correspondingly, the right-hand side will contain the desired expression for $\mathscr{T}(t)$.

Let H be a Hamiltonian consisting of two terms $H = A + B$, so that the ideal evolution is given by $\mathscr{E}(t) = e^{-it(A+B)}$. Consider simulating this Hamiltonian using an s-stage higher-order formula written in the canonical form $\mathscr{S}(t) = \mathscr{S}_s(t) \cdots \mathscr{S}_2(t)\mathscr{S}_1(t)$, where $\mathscr{S}_j(t) = e^{-itb_jB}e^{-ita_jA}$ is the jth-stage operator and $a_1, \ldots, a_s$ and $b_1, \ldots, b_s$ are real numbers. We adopt the convention $\prod_{l=1}^s \mathscr{S}_l(t) = \mathscr{S}_s(t)\mathscr{S}_{s-1}(t) \cdots \mathscr{S}_1(t)$ and let $b_0 = 0$.

We define $\mathscr{R}(t) := \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}(t) - (-iH)\mathscr{S}(t)$. Our goal is to obtain a concrete expression for $\mathscr{T}(t)$ satisfying $\mathscr{R}(t) = \mathscr{S}(t)\mathscr{T}(t)$. We have

$$\begin{aligned}\mathscr{R}(t) =&\frac{\mathrm{d}}{\mathrm{d}t}\Big[\prod_{j=1}^{s}\mathscr{S}_j(t)\Big] - (-i)(A+B)\prod_{j=1}^{s}\mathscr{S}_j(t)\\ =&\sum_{j=1}^{s}\Big(\prod_{l=j+1}^{s}\mathscr{S}_l(t)\Big)\Big(\mathscr{S}_j(t)(-ia_jA)+(-ib_jB)\mathscr{S}_j(t)\Big)\Big(\prod_{l=1}^{j-1}\mathscr{S}_l(t)\Big)\\ &-\sum_{j=1}^{s}\Big((-ia_jA)+(-ib_jB)\Big)\prod_{l=1}^{s}\mathscr{S}_l(t),\end{aligned} \tag{26}$$

where the second equality follows from the rule of differentiation and the fact that $\mathscr{S}(t)$ is at least first-order accurate, so $\sum_{j=1}^{s} a_j = \sum_{j=1}^{s} b_j$=1. We re-express the differences of operators as commutators to get

$$\mathscr{R}(t) = -i\sum_{j=1}^{s}\Big[\prod_{l=j}^{s}\mathscr{S}_l(t),\ a_jA + b_{j-1}B\Big]\prod_{l=1}^{j-1}\mathscr{S}_l(t). \tag{27}$$

Performing the commutation sequentially, we have

$$\mathscr{R}(t) = -i\sum_{j=1}^{s}\sum_{k=j}^{s}\Big(\prod_{l=k+1}^{s}\mathscr{S}_l(t)\big[\mathscr{S}_k(t),\ a_jA + b_{j-1}B\big]\prod_{l=j}^{k-1}\mathscr{S}_l(t)\Big)\prod_{l=1}^{j-1}\mathscr{S}_l(t). \tag{28}$$

To proceed, we interchange the order of summation, giving

$$\begin{aligned}\mathscr{R}(t) =&-i\sum_{k=1}^{s}\sum_{j=1}^{k}\prod_{l=k+1}^{s}\mathscr{S}_l(t)\big[\mathscr{S}_k(t),\ a_jA + b_{j-1}B\big]\prod_{l=1}^{k-1}\mathscr{S}_l(t)\\ =&-i\sum_{k=1}^{s}\prod_{l=k+1}^{s}\mathscr{S}_l(t)\big[\mathscr{S}_k(t),\ c_kA + d_{k-1}B\big]\prod_{l=1}^{k-1}\mathscr{S}_l(t),\end{aligned} \tag{29}$$

where we define

$$c_k := \sum_{j=1}^{k} a_j, \qquad d_k := \sum_{j=1}^{k} b_j. \tag{30}$$

Finally, we perform unitary conjugation to create $\mathscr{S}(t)$ on the left-hand side of (29). Specifically, we have

$$\begin{aligned}&-i\sum_{k=1}^{s}\prod_{l=k}^{s}\mathscr{S}_l(t)\cdot\big(c_kA + d_{k-1}B\big)\cdot\prod_{l=1}^{k-1}\mathscr{S}_l(t)\\ =&-i\mathscr{S}(t)\sum_{k=1}^{s}\prod_{l=k-1}^{1}\mathscr{S}_l^{\dagger}(t)\cdot\big(c_kA + d_{k-1}B\big)\cdot\prod_{l=1}^{k-1}\mathscr{S}_l(t)\end{aligned} \tag{31}$$

and

$$\begin{aligned}&-i\sum_{k=1}^{s}\prod_{l=k+1}^{s}\mathscr{S}_l(t)\cdot\big(c_kA + d_{k-1}B\big)\cdot\prod_{l=1}^{k}\mathscr{S}_l(t)\\ =&-i\mathscr{S}(t)\sum_{k=1}^{s}\prod_{l=k}^{1}\mathscr{S}_l^{\dagger}(t)\cdot\big(c_kA + d_{k-1}B\big)\cdot\prod_{l=1}^{k}\mathscr{S}_l(t).\end{aligned} \tag{32}$$

We have now established the following theorem.

Theorem 2 (Simplified local error representation). *Let $H = A + B$ be a Hamiltonian, so that the ideal evolution induced by H is $\mathscr{E}(t) = e^{-it(A+B)}$. Let $\mathscr{S}(t)$ be an s-stage product formula written in the canonical form*

$$\mathscr{S}(t) = \mathscr{S}_s(t) \cdots \mathscr{S}_2(t)\mathscr{S}_1(t) = \big(e^{-itb_sB}e^{-ita_sA}\big)\cdots\big(e^{-itb_2B}e^{-ita_2A}\big)\big(e^{-itb_1B}e^{-ita_1A}\big), \tag{33}$$

where $a_1, \ldots, a_s$ and $b_1, \ldots, b_s$ are real numbers, and $\mathscr{S}_j(t) = e^{-itb_jB}e^{-ita_jA}$ is the j-th stage operator. Then the product-formula error $\mathscr{S}(t) - \mathscr{E}(t)$ admits the integral representation

$$\mathscr{S}(t) - \mathscr{E}(t) = \int_0^t \mathscr{E}(t-\tau)\mathscr{R}(\tau)\mathrm{d}\tau, \tag{34}$$

where

$$\mathscr{R}(\tau) = \mathscr{S}(\tau)\mathscr{T}(\tau) \tag{35}$$

and

$$\begin{aligned}\mathscr{T}(\tau) = -i\sum_{k=1}^{s}\Bigg(&\prod_{l=k-1}^{1}\mathscr{S}_l^{\dagger}(\tau)\cdot\big(c_kA + d_{k-1}B\big)\cdot\prod_{l=1}^{k-1}\mathscr{S}_l(\tau)\\ &-\prod_{l=k}^{1}\mathscr{S}_l^{\dagger}(\tau)\cdot\big(c_kA + d_{k-1}B\big)\cdot\prod_{l=1}^{k}\mathscr{S}_l(\tau)\Bigg).\end{aligned} \tag{36}$$

Furthermore, if $\mathscr{S}(t)$ is a pth-order product formula, then

$$\mathscr{T}(\tau) = \int_0^{\tau}\mathrm{d}v\ \frac{\mathscr{T}^{(p)}(v)}{(p-1)!}(\tau - v)^{p-1} = p\int_0^1 \mathrm{d}x\ (1-x)^{p-1}\mathscr{T}^{(p)}(x\tau)\frac{\tau^p}{p!}. \tag{37}$$

Proof. Equation (36) follows from the discussion above. The integral representation (37) follows from Theorem 1 and Taylor's theorem with integral remainder. □

The local error representation developed by Descombes and Thalhammer [5, Theorem 1] is proved through a similar calculation as in (29), except that they use two additional rules for manipulating matrix exponentials: one for creating exponentials [5, Eq. (2.9a)] and the other for pushing matrix exponentials [5, Eq. (2.9b)]. Unfortunately, they overlooked a time-dependent term in their calculation when establishing the second rule. Furthermore, Descombes and Thalhammer's analysis relies on auxiliary functions defined recursively in terms of integrals denoted $\mathscr{I}_1$ and $\mathscr{I}_2$, whose combinatorial structure is hard to unravel. In contrast, our local error representation follows from a unitary conjugation trick that significantly simplifies the calculations. Therefore, we use our Theorem 2 in subsequent analysis of the product-formula algorithm.

IV. ADJOINT MAPPINGS AND ANALYSIS OF THE pTH-ORDER ALGORITHM

In this section, we give a detailed analysis of the pth-order product-formula algorithm for lattice simulation. We introduce the notion of adjoint mappings in Section IV A and use it to obtain a bound on the product-formula error in Section IV B.

A. Adjoint mappings

For any invertible matrix X, we define Ad_X to be the conjugation transformation given by

$$\mathrm{Ad}_X(Y) = XYX^{-1} \tag{38}$$

for any operator Y. Also for an arbitrary operator X, we define ad_X to be the commutator transformation, i.e.

$$\mathrm{ad}_X(Y) = \big[X, Y\big] = XY - YX \tag{39}$$

for any operator Y. These definitions are motivated by the notion of adjoint representation in the study of Lie groups and Lie algebras [6].

In the following proposition, we state a differentiation rule for Ad and ad, which will be useful when we compute the Taylor expansion of a multivariate function.

Proposition 1 (Differentiation rule)**.** *Let X be an operator and let $Y(t)$ be an operator-valued function that is infinitely differentiable. Then*

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\mathrm{Ad}_{e^{tX}}(Y(t))\right] = \mathrm{Ad}_{e^{tX}}(\mathrm{ad}_X(Y(t))) + \mathrm{Ad}_{e^{tX}}(Y'(t)). \tag{40}$$

Proof. The proof is a straightforward calculation. □

Corollary 1 (Higher-order differentiation rule)**.** *Let p be a positive integer, let X be an operator, and let $Y(t)$ be an operator-valued function that is infinitely differentiable. Then*

$$\frac{\mathrm{d}^p}{\mathrm{d}t^p}\left[\mathrm{Ad}_{e^{tX}}(Y(t))\right] = \sum_{j=0}^{p}\binom{p}{j}\mathrm{Ad}_{e^{tX}}\mathrm{ad}_X^j\left(Y^{(p-j)}(t)\right). \tag{41}$$

Proof. The claimed rule follows by Proposition 1 and the proof of the general Leibniz rule. □

In our analysis, a sequence of operators of the form

$$\begin{aligned}
&\mathrm{Ad}_{e^{\tau_1 X_{11}}}\cdots\mathrm{Ad}_{e^{\tau_1 X_{1\gamma_1}}}\mathrm{ad}_{Y_{11}}\cdots\mathrm{ad}_{Y_{1\delta_1}}\\
&\mathrm{Ad}_{e^{\tau_1 X_{21}}}\cdots\mathrm{Ad}_{e^{\tau_1 X_{2\gamma_2}}}\mathrm{ad}_{Y_{21}}\cdots\mathrm{ad}_{Y_{2\delta_2}}\\
&\cdots\\
&\mathrm{Ad}_{e^{\tau_m X_{m1}}}\cdots\mathrm{Ad}_{e^{\tau_m X_{m\gamma_m}}}\mathrm{ad}_{Y_{m1}}\cdots\mathrm{ad}_{Y_{m\delta_m}}(Z)
\end{aligned} \tag{42}$$

will be abbreviated as

$$\mathrm{Ad}_{\tau_1}^{\gamma_1}\mathrm{ad}^{\delta_1}\cdots\mathrm{Ad}_{\tau_m}^{\gamma_m}\mathrm{ad}^{\delta_m}(Z). \tag{43}$$

In other words, we omit the information about the operators and only keep track of the time variables $\tau_1,\ldots,\tau_m$. The advantage of this abbreviation is illustrated in the following proposition.

Proposition 2 (Differentiation rule for abbreviated adjoint representation)**.** *The following differentiation rule for the abbreviated adjoint representation holds:*

$$\begin{aligned}
&\frac{\partial^{w_1+\cdots+w_m}}{\partial\tau_1^{w_1}\cdots\partial\tau_m^{w_m}}\mathrm{Ad}_{\tau_1}^{\gamma_1}\mathrm{ad}^{\delta_1}\cdots\mathrm{Ad}_{\tau_m}^{\gamma_m}\mathrm{ad}^{\delta_\ell}(Z)\\
&=\sum_{v_{11}+\cdots+v_{1\gamma_1}=w_1}\binom{w_1}{v_{11}\ \cdots\ v_{1\gamma_1}}\left(\mathrm{Ad}_{\tau_1}\mathrm{ad}^{v_{11}}\right)\cdots\left(\mathrm{Ad}_{\tau_1}\mathrm{ad}^{v_{1\gamma_1}}\right)\mathrm{ad}^{\delta_1}\\
&\quad\cdots\\
&\quad\sum_{v_{m1}+\cdots+v_{m\gamma_1}=w_m}\binom{w_m}{v_{m1}\ \cdots\ v_{m\gamma_m}}\left(\mathrm{Ad}_{\tau_m}\mathrm{ad}^{v_{m1}}\right)\cdots\left(\mathrm{Ad}_{\tau_m}\mathrm{ad}^{v_{m\gamma_m}}\right)\mathrm{ad}^{\delta_m}(Z).
\end{aligned} \tag{44}$$

Proof. To prove the stated rule, it suffices to separate the time variables and prove that

$$\frac{\partial^{w_1}}{\partial\tau_1^{w_1}}\mathrm{Ad}_{\tau_1}^{\gamma_1}(Z) = \sum_{v_{11}+\cdots+v_{1\gamma_1}=w_1}\binom{w_1}{v_{11}\ \cdots\ v_{1\gamma_1}}\left(\mathrm{Ad}_{\tau_1}\mathrm{ad}^{v_{11}}\right)\cdots\left(\mathrm{Ad}_{\tau_1}\mathrm{ad}^{v_{1\gamma_1}}\right)(Z). \tag{45}$$

This follows by Corollary 1 and the proof of the multi-factor Leibniz rule. □

B. Error analysis of the pth-order algorithm

Suppose that we want to simulate a Hamiltonian H consisting of two terms $H = A + B$ for time t, so that the ideal evolution is given by $\mathscr{E}(t) = e^{-it(A+B)}$. As mentioned in Section II, a higher-order product formula may be represented in the canonical form

$$\mathscr{S}(t) = \mathscr{S}_s(t)\cdots\mathscr{S}_2(t)\mathscr{S}_1(t) = \big(e^{-itb_sB}e^{-ita_sA}\big)\cdots\big(e^{-itb_2B}e^{-ita_2A}\big)\big(e^{-itb_1B}e^{-ita_1A}\big), \tag{46}$$

where s is the number of stages and $a_1, \ldots, a_s, b_1, \ldots, b_s \in \mathbb{R}$. By Theorem 2, we know that the product-formula error $\mathscr{S}(t) - \mathscr{E}(t)$ admits the integral representation

$$\begin{aligned} \mathscr{S}(t) - \mathscr{E}(t) &= \int_0^t \mathscr{E}(t-\tau)\mathscr{R}(\tau)\mathrm{d}\tau, \\ \mathscr{R}(\tau) =& \mathscr{S}(\tau)\mathscr{T}(\tau), \end{aligned} \tag{47}$$

where

$$\begin{aligned} \mathscr{T}(\tau) = -i\sum_{k=1}^{s}\Bigg\{ &\prod_{l=k-1}^{1}\mathscr{S}_l^\dagger(\tau)\big(c_kA + d_{k-1}B\big)\prod_{l=1}^{k-1}\mathscr{S}_l(\tau) \\ &- \prod_{l=k}^{1}\mathscr{S}_l^\dagger(\tau)\big(c_kA + d_{k-1}B\big)\prod_{l=1}^{k}\mathscr{S}_l(\tau)\Bigg\}. \end{aligned} \tag{48}$$

Fix $1 \le k \le s$. We observe that the first term in (48) has the abbreviated adjoint representation

$$\mathrm{Ad}_\tau^{2(k-1)}\big(c_kA + d_{k-1}B\big). \tag{49}$$

To establish the scaling $O(t^{p+1})$, it suffices to show that the τ-dependence of $\mathscr{T}$ is $O(\tau^p)$. From Theorem 1, we know that terms of order $p-1$ or less will vanish, so we only need to compute the integral remainder of the Taylor expansion of each $\mathrm{Ad}_\tau^{2(k-1)}\big(c_kA + d_{k-1}B\big)$ at order p. In light of the chain rule, we apply the multivariate Taylor theorem and obtain the remainder

$$\begin{aligned} &p\int_0^1 \mathrm{d}s\ (1-s)^{p-1}\big[\mathrm{Ad}_\tau^{2(k-1)}\big(c_kA + d_{k-1}B\big)\big]^{(p)}\frac{\tau^p}{p!} \\ =&p\int_0^1 \mathrm{d}s\ (1-s)^{p-1} \\ &\sum_{w_1+\cdots+w_{2(k-1)}=p}\frac{\partial^p}{\partial\tau_1^{w_1}\cdots\partial\tau_{2(k-1)}^{w_{2(k-1)}}}\mathrm{Ad}_{\tau_1}\cdots\mathrm{Ad}_{\tau_{2(k-1)}}\big(c_kA + d_{k-1}B\big)\frac{\tau_1^{w_1}\cdots\tau_{2(k-1)}^{w_{2(k-1)}}}{w_1!\cdots w_{2(k-1)}!} \\ =&p\int_0^1 \mathrm{d}s\ (1-s)^{p-1} \\ &\sum_{w_1+\cdots+w_{2(k-1)}=p}\mathrm{Ad}_{\tau_1}\mathrm{ad}^{w_1}\cdots\mathrm{Ad}_{\tau_{2(k-1)}}\mathrm{ad}^{w_{2(k-1)}}\big(c_kA + d_{k-1}B\big)\frac{\tau_1^{w_1}\cdots\tau_{2(k-1)}^{w_{2(k-1)}}}{w_1!\cdots w_{2(k-1)}!}, \end{aligned} \tag{50}$$

where $\tau_1 = \cdots = \tau_{2(k-1)} = \tau$.

We assume that $\mathscr{S}(t)$ is an (s,p,u)-formula. To simulate an n-qubit lattice Hamiltonian $H = \sum_{j=1}^{n-1} H_{j,j+1}$, we instantiate

$$\begin{aligned} A = H_{\mathrm{odd}} = H_{1,2} + H_{3,4} + \cdots = \sum_{k=1}^{\frac{n}{2}} H_{2k-1,2k} \\ B = H_{\mathrm{even}} = H_{2,3} + H_{4,5} + \cdots = \sum_{k=1}^{\frac{n}{2}-1} H_{2k,2k+1}. \end{aligned} \tag{51}$$

We claim that

$$\left\|\mathrm{Ad}_{\tau_1}\mathrm{ad}^{w_1}\cdots\mathrm{Ad}_{\tau_{2(k-1)}}\mathrm{ad}^{w_{2(k-1)}}\left(c_kA+d_{k-1}B\right)\right\|\leq n(2k-1)^p(ku)(2u)^p. \tag{52}$$

To see this, first note that we have operator $c_kA+d_{k-1}B$ in the inner-most layer, which contains at most n terms, each of which has spectral norm at most ku. Now, we fix a particular term $c_kH_{2\eta-1,2\eta}$ and study the abbreviated adjoint representation in (52). The spectral norm will increase by a factor of $2u$ every time an ad is composed, and will remain the same if an Ad is composed. The total number of ad's is p, explaining the factor $(2u)^p$ in (52).

The justification of the factor $(2k-1)^p$ is more difficult. At the beginning, we have the operator $c_kH_{2\eta-1,2\eta}$. When the first ad is applied, we have

$$\left[-ib_{k-1}H_{\mathrm{even}},c_kH_{2\eta-1,2\eta}\right], \tag{53}$$

which only contains two nonzero commutators

$$\left[-ib_{k-1}H_{2\eta-2,2\eta-1},c_kH_{2\eta-1,2\eta}\right],\qquad\left[-ib_{k-1}H_{2\eta,2\eta+1},c_kH_{2\eta-1,2\eta}\right]. \tag{54}$$

We see that the support of the operator is enlarged from qubits $2\eta-1,2\eta$ to $2\eta-2,2\eta-1,2\eta,2\eta+1$. The next $w_{2(k-1)}-1$ ad's all represent commutators with $-ib_{k-1}H_{\mathrm{even}}$. Therefore, the support of the operator will remain unchanged if more ad's are composed. When the next Ad is composed, we break the exponential of H_{even} into product of elementary exponentials of $H_{2\eta,2\eta+1}$, and cancel as many terms as possible in pairs. This does not enlarge the support either.

However, the next ad represents a commutator with $-ia_{k-1}H_{\mathrm{odd}}$. After cancellation, the support of the operator is enlarged to

$$2\eta-3,2\eta-2,2\eta-1,2\eta,2\eta+1,2\eta+2. \tag{55}$$

Following this argument, we find that the support of operators increases by two every time an Ad is composed. The total number of Ad's is $2(k-1)$, so the support of the last operator will be at most $4k-2$. This upper bounds the number of nonzero nested commutators by $(2k-1)^p$.

The analysis is similar when the term in the inner-most layer of (52) is $d_{k-1}H_{2\eta,2\eta+1}$. Therefore, the remainder is upper bounded by

$$\begin{aligned}&p\int_0^1\mathrm{d}s\ (1-s)^{p-1}\sum_{w_1+\cdots+w_{2(k-1)}=p}n(2k-1)^p(ku)(2u)^p\frac{\tau^p}{w_1!\cdots w_{2(k-1)}!}\\&=n(2k-1)^p(ku)(2u)^p(2k-2)^p\frac{\tau^p}{p!}.\end{aligned} \tag{56}$$

A summation over $1\leq k\leq s$ and an integration $\int_0^t\mathrm{d}\tau$ give

$$\sum_{k=1}^{s}n(2k-1)^p(ku)(2u)^p(2k-2)^p\frac{t^{p+1}}{(p+1)!}. \tag{57}$$

Similarly, we find that the second term in (48) can be upper bounded by

$$\sum_{k=1}^{s}n(2k+1)^p(ku)(2u)^p(2k)^p\frac{t^{p+1}}{(p+1)!}. \tag{58}$$

Therefore, the product-formula error scales like $O(nt^{p+1})$ assuming that s, p, and u are constant.

V. TIME-DEPENDENT PRODUCT FORMULAS AND LOCAL ERROR ANALYSIS

In this appendix, we discuss time-dependent product formulas and their local error analysis in detail. In Section V A, we introduce canonical formulas for time-dependent Hamiltonian simulation and study their order conditions. We then analyze the time-dependent local error structure in Section V B.

A. Time-dependent canonical formulas and order conditions

Let $H(t)$ be a Hamiltonian that depends on the time variable t. We can express the evolution under $H(t)$ for time t as

$$\mathscr{E}_{\mathcal{T}}(t) := \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}v\ H(v)\right), \tag{59}$$

where $\mathscr{E}_{\mathcal{T}}(t)$ denotes the time-ordered exponential. The operator $\mathscr{E}_{\mathcal{T}}(t)$ is unitary and satisfies the differentiation rule

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathscr{E}_{\mathcal{T}}(t) = -iH(t)\mathscr{E}_{\mathcal{T}}(t). \tag{60}$$

Throughout this section, we assume that the Hamiltonian $H(t)$ and its terms are infinitely differentiable with respect to t, which ensures that a product formula can approximate the ideal evolution to the stated order. The infinite differentiability of $H(t)$ may be relaxed [7], but we impose this assumption to make the presentation cleaner.

Assuming that $H(t)$ is infinitely differentiable, we have the following rule for computing higher-order derivatives of $\mathscr{E}_{\mathcal{T}}(t)$.

Lemma 2 (Higher-order derivatives of $\mathscr{E}_{\mathcal{T}}(t)$ [7, Lemma 1])**.** *Let $H(t)$ be a time-dependent Hamiltonian that is infinitely differentiable. Then the evolution operator $\mathscr{E}_{\mathcal{T}}(t) = \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}v\ H(v)\right)$ is also infinitely differentiable and its derivatives are*

$$\mathscr{E}_{\mathcal{T}}^{(j)}(t) = T_j(t)\mathscr{E}_{\mathcal{T}}(t), \tag{61}$$

where the $T_j(t)$ are specified by the recurrence

$$\begin{aligned} T_0 &= I, \\ T_j(t) &= -iT_{j-1}(t)H(t) + \frac{\mathrm{d}}{\mathrm{d}t}T_{j-1}(t). \end{aligned} \tag{62}$$

We now show that $T_j(t)$ satisfies the following higher-order recursive formula.

Lemma 3 (Recursive formula for $T_j(t)$)**.** *For all $j \in \mathbb{N}$, the operator-valued function $T_j(t)$ defined in (62) satisfies*

$$T_{j+1}(t) = -i\sum_{k=0}^{j}\binom{j}{k}H^{(k)}(t)T_{j-k}(t). \tag{63}$$

Proof. We first prove by induction that

$$\mathscr{E}_{\mathcal{T}}^{(j+1)}(t) = -i\sum_{k=0}^{j}\binom{j}{k}H^{(k)}(t)\mathscr{E}_{\mathcal{T}}^{(j-k)}(t). \tag{64}$$

For the base case $j = 0$, the claimed equality reduces to

$$\mathscr{E}_{\mathcal{T}}^{(1)}(t) = -iH^{(0)}(t)\mathscr{E}_{\mathcal{T}}^{(0)}(t), \tag{65}$$

which follows trivially from (60).

Now suppose that

$$\mathscr{E}_{\mathcal{T}}^{(j)}(t) = -i\sum_{k=0}^{j-1}\binom{j-1}{k}H^{(k)}(t)\mathscr{E}_{\mathcal{T}}^{(j-1-k)}(t). \tag{66}$$

Differentiating both sides of the above equation, we have

$$\begin{aligned}
\mathscr{E}_{\mathcal{T}}^{(j+1)}(t) =& \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{E}_{\mathcal{T}}^{(j)}(t) \\
=& -i\frac{\mathrm{d}}{\mathrm{d}t}\sum_{k=0}^{j-1}\binom{j-1}{k}H^{(k)}(t)\mathscr{E}_{\mathcal{T}}^{(j-1-k)}(t) \\
=& -i\sum_{k=0}^{j-1}\binom{j-1}{k}H^{(k+1)}(t)\mathscr{E}_{\mathcal{T}}^{(j-1-k)}(t) - i\sum_{k=0}^{j-1}\binom{j-1}{k}H^{(k)}(t)\mathscr{E}_{\mathcal{T}}^{(j-k)}(t) \\
=& -i\sum_{k=1}^{j}\binom{j-1}{k-1}H^{(k)}(t)\mathscr{E}_{\mathcal{T}}^{(j-k)}(t) - i\sum_{k=0}^{j-1}\binom{j-1}{k}H^{(k)}(t)\mathscr{E}_{\mathcal{T}}^{(j-k)}(t) \\
=& -i\sum_{k=0}^{j}\binom{j}{k}H^{(k)}(t)T_{j-k}(t).
\end{aligned} \tag{67}$$

Thus (64) follows by induction.

We now invoke Lemma 2 to find

$$T_{j+1}(t)\mathscr{E}_{\mathcal{T}}(t) = \sum_{k=0}^{j}\binom{j}{k}[-iH^{(k)}(t)]T_{j-k}(t)\mathscr{E}_{\mathcal{T}}(t). \tag{68}$$

Canceling the unitary operator $\mathscr{E}_{\mathcal{T}}(t)$ proves the claimed recursive formula for $T_j(t)$. □

Now let $H(t)$ be a time-dependent Hamiltonian consisting of two terms $H(t) = A(t)+B(t)$, where $A(t)$ and $B(t)$ are Hermitian operators that are infinitely differentiable with respect to t. We simulate the evolution under $H(t)$ using a product formula of the form

$$\begin{aligned}
\mathscr{S}_{\mathcal{T}}(t) =& \mathscr{S}_{\mathcal{T},s}(t)\cdots\mathscr{S}_{\mathcal{T},2}(t)\mathscr{S}_{\mathcal{T},1}(t) \\
=& \left(e^{-itb_sB(t\beta_s)}e^{-ita_sA(t\alpha_s)}\right)\cdots\left(e^{-itb_2B(t\beta_2)}e^{-ita_2A(t\alpha_2)}\right)\left(e^{-itb_1B(t\beta_1)}e^{-ita_1A(t\alpha_1)}\right),
\end{aligned} \tag{69}$$

where $a_k, b_k, \alpha_k, \beta_k$ are real numbers. We call $\mathscr{S}_{\mathcal{T}}(t)$ a time-dependent canonical product formula with s stages, where $\mathscr{S}_{\mathcal{T},k}(t) = e^{-itb_kB(t\beta_k)}e^{-ita_kA(t\alpha_k)}$ denotes the k-th stage operator for $k = 1,\dots,s$. Intuitively, this formula samples the Hamiltonian at times $t\beta_k$, $t\alpha_k$ and applies a time-independent product formula to approximate the ideal evolution.

We say that $\mathscr{S}_{\mathcal{T}}(t)$ is a pth-order product formula if

$$\mathscr{S}_{\mathcal{T}}(t) = \mathscr{E}_{\mathcal{T}}(t) + O(t^{p+1}). \tag{70}$$

Using Lemma 1, we find the order condition $\mathscr{S}_{\mathcal{T}}^{(j)}(0) = \mathscr{E}_{\mathcal{T}}^{(j)}(0)$ for $0 \le j \le p$, which is equivalent to

$$\mathscr{S}_{\mathcal{T}}^{(j)}(0) = T_j(0) \tag{71}$$

by Lemma 2.

Let $\mathscr{R}_{\mathcal{T}}(t)$ be an operator-valued function defined as

$$\mathscr{R}_{\mathcal{T}}(t) := \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}_{\mathcal{T}}(t) + iH(t)\mathscr{S}_{\mathcal{T}}(t). \tag{72}$$

Using the variation-of-parameters formula [8, Theorem 4.9], $\mathscr{R}_{\mathcal{T}}(t)$ facilitates an integral representation of the product-formula error:

$$\mathscr{S}_{\mathcal{T}}(t) - \mathscr{E}_{\mathcal{T}}(t) = \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}v\ H(v)\right)\int_0^t \mathrm{d}\tau_1\ \exp_{\mathcal{T}}^{\dagger}\left(-i\int_0^{\tau_1}\mathrm{d}v\ H(v)\right)\mathscr{R}_{\mathcal{T}}(\tau_1). \tag{73}$$

We claim that the order condition

$$\mathscr{S}_{\mathcal{T}}^{(j)}(0) = T_j(0), \quad 0 \le j \le p \tag{74}$$

is equivalent to

$$\mathscr{R}_{\mathcal{T}}^{(j)}(0) = 0, \quad 0 \le j \le p-1. \tag{75}$$

To see this, first suppose that (74) holds. For $0 \le j \le p-1$, we have

$$\begin{aligned}\mathscr{R}_{\mathcal{T}}^{(j)}(0) &= \mathscr{S}_{\mathcal{T}}^{(j+1)}(0) - \sum_{k=0}^{j} \binom{j}{k} [-iH^{(k)}(0)] \mathscr{S}_{\mathcal{T}}^{(j-k)}(0) \\ &= T_{j+1}(0) - \sum_{k=0}^{j} \binom{j}{k} [-iH^{(k)}(0)] T_{j-k}(0) = 0,\end{aligned} \tag{76}$$

where the first equality follows from the definition of $\mathscr{R}_{\mathcal{T}}$ and the general Leibniz rule; the second equality follows from the order condition (74); and the last equality follows from Lemma 3. Conversely, suppose that (75) holds. Using the general Leibniz rule, we have

$$\mathscr{S}_{\mathcal{T}}^{(j+1)}(0) = \sum_{k=0}^{j} \binom{j}{k} [-iH^{(k)}(0)] \mathscr{S}_{\mathcal{T}}^{(j-k)}(0) \tag{77}$$

for $0 \le j \le p-1$. But we know from Lemma 3 that T_{j+1} satisfies the same recursive formula, with the base case

$$\mathscr{S}_{\mathcal{T}}(0) = I = T_0. \tag{78}$$

An inductive argument gives (74).

To get the correct scaling of the product-formula error in both n and t, we introduce another operator-valued function $\mathscr{T}_{\mathcal{T}}$ defined as

$$\mathscr{T}_{\mathcal{T}} := \mathscr{S}_{\mathcal{T}}^{\dagger} \mathscr{R}_{\mathcal{T}}. \tag{79}$$

The order condition for $\mathscr{T}_{\mathcal{T}}$ is

$$\mathscr{T}_{\mathcal{T}}^{(j)}(0) = 0, \quad 0 \le j \le p-1, \tag{80}$$

which follows in the same way as in Section II. We have now established several order conditions for time-dependent Hamiltonian simulation, which we summarize as follows.

Theorem 3 (Order conditions for time-dependent canonical product formulas)**.** *Let $H(t)$ be a time-dependent Hamiltonian consisting of two terms $H(t) = A(t) + B(t)$, such that both $A(t)$ and $B(t)$ are infinitely differentiable. Let $p \ge 1$ be an integer and let $\mathscr{S}_{\mathcal{T}}(t)$ be a time-dependent canonical product formula for $H(t)$. The following four conditions are equivalent.*

1. *$\mathscr{S}_{\mathcal{T}}(t) = \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}v\ H(v)\right) + O(t^{p+1})$.*

2. *$\mathscr{S}_{\mathcal{T}}^{(j)}(0) = T_j(0)$ for all $0 \le j \le p$. Here, $T_j(t)$ are defined recursively as*

$$\begin{aligned}T_0 &= I, \\ T_j(t) &= T_{j-1}(t)\big[-iH(t)\big] + \frac{\mathrm{d}}{\mathrm{d}t} T_{j-1}(t).\end{aligned} \tag{81}$$

3. *There is some infinitely differentiable operator-valued function $\mathscr{R}_{\mathcal{T}}(t)$ with $\mathscr{R}_{\mathcal{T}}^{(j)}(0) = 0$ for all $0 \le j \le p-1$, such that*

$$\begin{aligned}\mathscr{S}_{\mathcal{T}}(t) =& \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}v\ H(v)\right) \\ &+ \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}v\ H(v)\right) \int_0^t \mathrm{d}\tau_1\ \exp_{\mathcal{T}}^{\dagger}\left(-i\int_0^{\tau_1} \mathrm{d}v\ H(v)\right) \mathscr{R}_{\mathcal{T}}(\tau_1).\end{aligned} \tag{82}$$

4. There is some infinitely differentiable operator-valued function $\mathscr{T}_{\mathcal{T}}(t)$ with $\mathscr{T}_{\mathcal{T}}^{(j)}(0) = 0$ for all $0 \le j \le p-1$, such that

$$\begin{aligned}\mathscr{S}_{\mathcal{T}}(t) =& \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}v\ H(v)\right)\\ &+ \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}v\ H(v)\right)\int_0^t \mathrm{d}\tau_1\ \exp_{\mathcal{T}}^{\dagger}\left(-i\int_0^{\tau_1}\mathrm{d}v\ H(v)\right)\mathscr{S}_{\mathcal{T}}(\tau_1)\mathscr{T}_{\mathcal{T}}(\tau_1).\end{aligned} \tag{83}$$

Furthermore, $\mathscr{R}_{\mathcal{T}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}_{\mathcal{T}}(t) - [-iH(t)]\mathscr{S}_{\mathcal{T}}(t)$ and $\mathscr{T}_{\mathcal{T}}(t) = \mathscr{S}_{\mathcal{T}}(t)^{\dagger}\mathscr{R}_{\mathcal{T}}(t)$ are uniquely determined.

B. Time-dependent local error representation

We now derive a local error representation for time-dependent Hamiltonian simulation. Let $H(t)$ be a time-dependent Hamiltonian consisting of two terms $H(t) = A(t) + B(t)$. We assume that both $A(t)$ and $B(t)$ are infinitely differentiable, although this assumption can be relaxed using techniques from [7]. The ideal evolution under $H(t)$ for time t is given by the time-ordered exponential

$$\mathscr{E}_{\mathcal{T}}(t) = \exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}v\ H(v)\right), \tag{84}$$

which we simulate using a time-dependent canonical product formula

$$\begin{aligned}\mathscr{S}_{\mathcal{T}}(t) =& \mathscr{S}_{\mathcal{T},s}(t)\cdots\mathscr{S}_{\mathcal{T},2}(t)\mathscr{S}_{\mathcal{T},1}(t)\\ =& \left(e^{-itb_sB(t\beta_s)}e^{-ita_sA(t\alpha_s)}\right)\cdots\left(e^{-itb_2B(t\beta_2)}e^{-ita_2A(t\alpha_2)}\right)\left(e^{-itb_1B(t\beta_1)}e^{-ita_1A(t\alpha_1)}\right).\end{aligned} \tag{85}$$

We know from Theorem 3 that the product-formula error admits an integral representation

$$\mathscr{S}_{\mathcal{T}}(t) = \mathscr{E}_{\mathcal{T}}(t) + \mathscr{E}_{\mathcal{T}}(t)\int_0^t \mathrm{d}\tau_1\ \mathscr{E}_{\mathcal{T}}^{\dagger}(\tau_1)\mathscr{R}_{\mathcal{T}}(\tau_1), \tag{86}$$

where $\mathscr{R}_{\mathcal{T}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}_{\mathcal{T}}(t) - [-iH(t)]\mathscr{S}_{\mathcal{T}}(t)$. A direct Taylor expansion of $\mathscr{R}_{\mathcal{T}}(t)$ will give the correct error scaling of t, but cannot easily be used to show the correct n-dependence. Instead, we consider an expansion of the operator $\mathscr{T}_{\mathcal{T}}(t) = \mathscr{S}_{\mathcal{T}}(t)^{\dagger}\mathscr{R}_{\mathcal{T}}(t)$. To this end, we compute $\mathscr{R}_{\mathcal{T}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}_{\mathcal{T}}(t) - (-iH)\mathscr{S}_{\mathcal{T}}(t)$ explicitly. We then perform unitary conjugation to create $\mathscr{S}_{\mathcal{T}}(t)$ on the left-hand side of $\mathscr{R}_{\mathcal{T}}(t)$. Correspondingly, the right-hand side will contain the desired expression for $\mathscr{T}_{\mathcal{T}}(t)$.

The following lemma is useful in our analysis.

Lemma 4 (Chain rule for matrix exponentiation [9, Eq.(29)], [10, Page 181])**.** *Let $G(x)$ be an operator-valued function of $x \in \mathbb{R}$ that is infinitely differentiable. Then the derivative $\frac{\mathrm{d}e^{G(x)}}{\mathrm{d}x}$ can be expressed as*

$$\frac{\mathrm{d}e^{G(x)}}{\mathrm{d}x} = \int_0^1 e^{yG(x)}\frac{\mathrm{d}G(x)}{\mathrm{d}x}e^{(1-y)G(x)}\mathrm{d}y = \int_0^1 e^{(1-z)G(x)}\frac{\mathrm{d}G(x)}{\mathrm{d}x}e^{zG(x)}\mathrm{d}z. \tag{87}$$

If we further define operator-valued functions

$$\begin{aligned}\mathscr{I}_{\mathcal{T},\mathrm{L}}\big(G(x),x\big) &= \int_0^1 \mathrm{d}y\ e^{yG(x)}G'(x)e^{-yG(x)}\\ \mathscr{I}_{\mathcal{T},\mathrm{R}}\big(G(x),x\big) &= \int_0^1 \mathrm{d}y\ e^{-yG(x)}G'(x)e^{yG(x)},\end{aligned} \tag{88}$$

then the chain rule can be succinctly expressed as

$$\frac{\mathrm{d}e^{G(x)}}{\mathrm{d}x} = \mathscr{I}_{\mathcal{T},\mathrm{L}}\big(G(x),x\big)e^{G(x)} = e^{G(x)}\mathscr{I}_{\mathcal{T},\mathrm{R}}\big(G(x),x\big). \tag{89}$$

We now compute

$$
\begin{aligned}
\mathscr{R}_{\mathcal{T}}(t) =&\frac{\mathrm{d}}{\mathrm{d}t}\mathscr{S}_{\mathcal{T}}(t)-[-iH(t)]\mathscr{S}_{\mathcal{T}}(t)\\
=&\frac{\mathrm{d}}{\mathrm{d}t}\left[\prod_{j=1}^{s}\mathscr{S}_{\mathcal{T},j}(t)\right]+iH(t)\prod_{j=1}^{s}\mathscr{S}_{\mathcal{T},j}(t)\\
=&\sum_{j=1}^{s}\left(\prod_{l=j+1}^{s}\mathscr{S}_{\mathcal{T},l}(t)\right)\Big(\mathscr{S}_{\mathcal{T},j}(t)\mathscr{I}_{\mathcal{T},\mathrm{R}}\big(-ita_jA(t\alpha_j),t\big)\\
&\qquad+\mathscr{I}_{\mathcal{T},\mathrm{L}}\big(-itb_jB(t\beta_j),t\big)\mathscr{S}_{\mathcal{T},j}(t)\Big)\left(\prod_{l=1}^{j-1}\mathscr{S}_{\mathcal{T},l}(t)\right)\\
&+iH(t)\prod_{l=1}^{s}\mathscr{S}_{\mathcal{T},l}(t),
\end{aligned}
\tag{90}
$$

where we have used the chain rule in the last equality. To proceed, we perform unitary conjugation to create the time-dependent product formula on the left-hand side as

$$
\begin{aligned}
&\sum_{j=1}^{s}\left(\prod_{l=j+1}^{s}\mathscr{S}_{\mathcal{T},l}(t)\right)\Big(\mathscr{S}_{\mathcal{T},j}(t)\mathscr{I}_{\mathcal{T},\mathrm{R}}\big(-ita_jA(t\alpha_j),t\big)\\
&\qquad+\mathscr{I}_{\mathcal{T},\mathrm{L}}\big(-itb_jB(t\beta_j),t\big)\mathscr{S}_{\mathcal{T},j}(t)\Big)\left(\prod_{l=1}^{j-1}\mathscr{S}_{\mathcal{T},l}(t)\right)\\
=&\mathscr{S}_{\mathcal{T}}(t)\sum_{j=1}^{s}\left(\prod_{l=j-1}^{1}\mathscr{S}^{\dagger}_{\mathcal{T},l}(t)\right)\mathscr{I}_{\mathcal{T},\mathrm{R}}\big(-ita_jA(t\alpha_j),t\big)\left(\prod_{l=1}^{j-1}\mathscr{S}_{\mathcal{T},l}(t)\right)\\
&+\mathscr{S}_{\mathcal{T}}(t)\sum_{j=1}^{s}\left(\prod_{l=j}^{1}\mathscr{S}^{\dagger}_{\mathcal{T},l}(t)\right)\mathscr{I}_{\mathcal{T},\mathrm{L}}\big(-itb_jB(t\beta_j),t\big)\left(\prod_{l=1}^{j}\mathscr{S}_{\mathcal{T},l}(t)\right)
\end{aligned}
\tag{91}
$$

and

$$
\begin{aligned}
&iH(t)\prod_{l=1}^{s}\mathscr{S}_{\mathcal{T},l}(t)\\
=&\mathscr{S}_{\mathcal{T}}(t)\prod_{l=s}^{1}\mathscr{S}_{\mathcal{T},l}(t)\big[iH(t)\big]\prod_{l=1}^{s}\mathscr{S}_{\mathcal{T},l}(t).
\end{aligned}
\tag{92}
$$

We have therefore established the following.

Theorem 4 (Time-dependent local error representation)**.** *theorem Let $H(t)=A(t)+B(t)$ be a time-dependent Hamiltonian with $A(t)$ and $B(t)$ infinitely differentiable, so that the ideal evolution under $H(t)$ for time t is given by $\mathscr{E}_{\mathcal{T}}(t)=\exp_{\mathcal{T}}\left(-i\int_0^t \mathrm{d}v\ H(v)\right)$. Let $\mathscr{S}_{\mathcal{T}}(t)$ be a time-dependent s-stage formula written in the canonical form*

$$
\begin{aligned}
\mathscr{S}_{\mathcal{T}}(t)=&\mathscr{S}_{\mathcal{T},s}(t)\cdots\mathscr{S}_{\mathcal{T},2}(t)\mathscr{S}_{\mathcal{T},1}(t)\\
=&\big(e^{-itb_sB(t\beta_s)}e^{-ita_sA(t\alpha_s)}\big)\cdots\big(e^{-itb_2B(t\beta_2)}e^{-ita_2A(t\alpha_2)}\big)\big(e^{-itb_1B(t\beta_1)}e^{-ita_1A(t\alpha_1)}\big),
\end{aligned}
\tag{93}
$$

where a_k,b_k,α_k,β_k are real numbers and $\mathscr{S}_{\mathcal{T},k}(t)=e^{-itb_kB(t\beta_k)}e^{-ita_kA(t\alpha_k)}$ is the k-th stage operator for $k\in\{1,\dots,s\}$. Then the product-formula error $\mathscr{S}_{\mathcal{T}}(t)-\mathscr{E}_{\mathcal{T}}(t)$ admits the integral representation

$$
\begin{aligned}
\mathscr{S}_{\mathcal{T}}(t)-\mathscr{E}_{\mathcal{T}}(t)=&\int_0^t\mathscr{E}_{\mathcal{T}}(t-\tau)\mathscr{R}_{\mathcal{T}}(\tau)\mathrm{d}\tau,\\
\mathscr{R}_{\mathcal{T}}(\tau)=&\mathscr{S}_{\mathcal{T}}(\tau)\mathscr{T}_{\mathcal{T}}(\tau),
\end{aligned}
\tag{94}
$$

where

$$\begin{aligned}\mathscr{T}_{\mathcal{T}}(\tau) = \sum_{j=1}^{s}\Bigg\{&\bigg(\prod_{l=j-1}^{1}\mathscr{S}_{\mathcal{T},l}^{\dagger}(t)\bigg)\mathscr{I}_{\mathcal{T},\mathrm{R}}\big(-ita_jA(t\alpha_j),t\big)\bigg(\prod_{l=1}^{j-1}\mathscr{S}_{\mathcal{T},l}(t)\bigg)\\ &-\bigg(\prod_{l=j}^{1}\mathscr{S}_{\mathcal{T},l}^{\dagger}(t)\bigg)\mathscr{I}_{\mathcal{T},\mathrm{L}}\big(-itb_jB(t\beta_j),t\big)\bigg(\prod_{l=1}^{j}\mathscr{S}_{\mathcal{T},l}(t)\bigg)\Bigg\}\\ &+\prod_{l=s}^{1}\mathscr{S}_{\mathcal{T},l}^{\dagger}(t)\big[iH(t)\big]\prod_{l=1}^{s}\mathscr{S}_{\mathcal{T},l}(t)\end{aligned} \tag{95}$$

and

$$\begin{aligned}\mathscr{I}_{\mathcal{T},\mathrm{L}}\big(G(x),x\big) &= \int_0^1 \mathrm{d}y\ e^{yG(x)}G'(x)e^{-yG(x)}\\ \mathscr{I}_{\mathcal{T},\mathrm{R}}\big(G(x),x\big) &= \int_0^1 \mathrm{d}y\ e^{-yG(x)}G'(x)e^{yG(x)}.\end{aligned} \tag{96}$$

Furthermore, if $\mathscr{S}_{\mathcal{T}}(t)$ is a time-dependent pth-order formula, then

$$\mathscr{T}_{\mathcal{T}}(\tau) = p\int_0^1 \mathrm{d}x\ (1-x)^{p-1}\mathscr{T}_{\mathcal{T}}^{(p)}(x\tau)\frac{\tau^p}{p!}. \tag{97}$$

Here (97) follows from the order conditions and Taylor's theorem with integral remainder as in Theorem 2.

VI. EMPIRICAL PERFORMANCE

The product-formula algorithm is the simplest approach to digital quantum simulation and its implementation does not require any ancilla qubits. We have shown that this algorithm can simulate a lattice Hamiltonian with nearly optimal gate complexity, and we established the ordering robustness property for the first-order algorithm. Recently, Haah, Hastings, Kothari, and Low (HHKL) proposed a new algorithm motivated by the Lieb-Robinson bounds, which also has nearly optimal complexity for lattice simulation and is ancilla-free if its each block is simulated by product formulas [11]. In this section, we numerically compare the empirical gate complexity of the product-formula algorithm and HHKL. We also consider the empirical performance of product formulas with respect to different orderings of lattice terms.

For concreteness, we consider a one-dimensional nearest-neighbor Heisenberg model with a random magnetic field. Its Hamiltonian has the form

$$H = \sum_{j=1}^{n-1}(\vec{\sigma}_j\cdot\vec{\sigma}_{j+1} + h_j\sigma_j^z) \tag{98}$$

with coefficients $h_j \in [-1,1]$ chosen uniformly at random, where $\vec{\sigma}_j = (\sigma_j^x,\sigma_j^y,\sigma_j^z)$ denotes a vector of Pauli x, y, and z matrices on qubit j. This is a widely studied model in condensed matter physics, whose simulation is beyond the reach of current classical computers for all but the smallest systems. Following [12], we set accuracy $\epsilon = 10^{-3}$ and choose the simulation time to be the same as the system size (i.e., $t = n$). To simplify the numerical implementation, we consider open boundary conditions, although our analysis can also be generalized to handle periodic conditions as described in [2]. This Hamiltonian has the form $H = 4\sum_{j=1}^{n-1}H_{j,j+1}$ with $H_{j,j+1} = (\vec{\sigma}_j\cdot\vec{\sigma}_{j+1} + h_j\sigma_j^z)/4$, so that $\max_j\|H_{j,j+1}\| \le 1$. Thus we normalize our Hamiltonian by a factor of 4 and simulate for time $4t$. We estimate the empirical gate complexity of product formulas as in [12].

In HHKL, the entire evolution is decomposed into $m/2$ blocks of evolutions on ℓ qubits and $m/2$ blocks of evolutions on 2ℓ qubits, each for time $t_\square$. We choose $\ell = 7$ to be constant and use the fitted data of [11] to obtain an error contribution of

$$0.175\bigg(\frac{7.9t_\square}{\ell+0.95}\bigg)^{\ell+0.95} = \frac{\epsilon}{3m}, \tag{99}$$

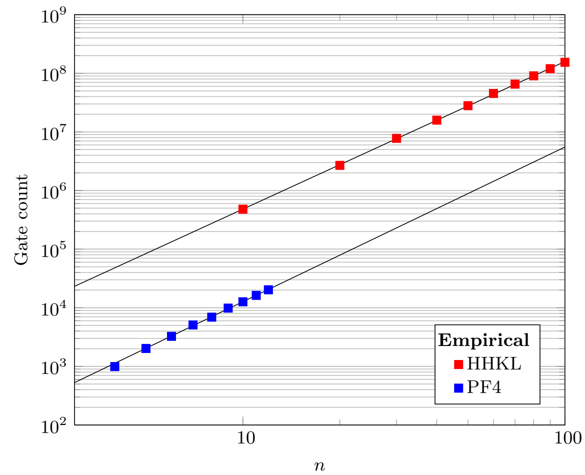

FIG. 1: Comparison of the empirical gate count between the HHKL algorithm with each block simulated by the fourth-order product formula and the (pure) fourth-order product-formula algorithm. Error bars are omitted as they are negligibly small on the plot. Straight lines show power-law fits to the data.

where the number of blocks is

$$m = \frac{8tn}{t_{\Box}\ell}. \tag{100}$$

We can therefore simultaneously solve for the number of blocks m and the evolution time per block $t_{\Box}$. We then use product formulas to simulate each block of size $\ell = 7$ or $2\ell = 14$ for time $t_{\Box}$ with accuracy $\epsilon_{\Box} = \epsilon/3m$. We choose the fourth-order product formula since it has the minimum gate count in practice for simulating the Heisenberg model of size up to 30 [12, Figure 3]. For a fair comparison, we also compute the empirical gate count of the pure fourth-order product-formula algorithm, whose performance is not too much worse than the best product formula for n up to 300 [12, Figure 3].

Figure 1 shows the resulting gate complexity for HHKL and the pure product-formula algorithm. Fitting the data, we obtain

$$g_{\mathrm{HHKL}} = 1461.453n^{2.518}, \qquad g_{\mathrm{PF}} = 29.093n^{2.639}. \tag{101}$$

We find that, while the asymptotic scaling of HHKL is better, the product-formula approach has a significantly better constant prefactor. Indeed, the HHKL algorithm introduces extra negative terms in the Hamiltonian to compensate for the error of the Lieb-Robinson decomposition and then simulates each block using product formulas [11], whereas the pure product-formula algorithm simulates the original lattice Hamiltonian with no overhead. Therefore, even though both algorithms are ancilla-free, the pure product-formula approach seems more desirable for near-term simulation.

To better understand the ordering robustness property, we also compare the empirical values of r for the first-order product-formula algorithm by ordering terms in the even-odd pattern and the X-Y-Z pattern of [12]. Figure 2 shows the resulting data, from which we estimate

$$r_{\text{even-odd}} = 586.816n^{1.942}, \qquad r_{\text{X-Y-Z}} = 668.139n^{2.507}. \tag{102}$$

Both are consistent with the claimed upper bound $r = O(nt^2) = O(n^3)$ for lattice simulation, but the even-odd ordering of terms gives better performance in practice.

[1] S. P. Jordan, K. S. M. Lee, and J. Preskill, Science **336**, 1130 (2012), arXiv:1111.3633.
[2] A. Childs and Y. Su, "Nearly optimal lattice simulation by product formulas (Main Paper)," (2019).
[3] D. Wecker, B. Bauer, B. K. Clark, M. B. Hastings, and M. Troyer, Physical Review A **90**, 022305 (2014), arXiv:1312.1695.

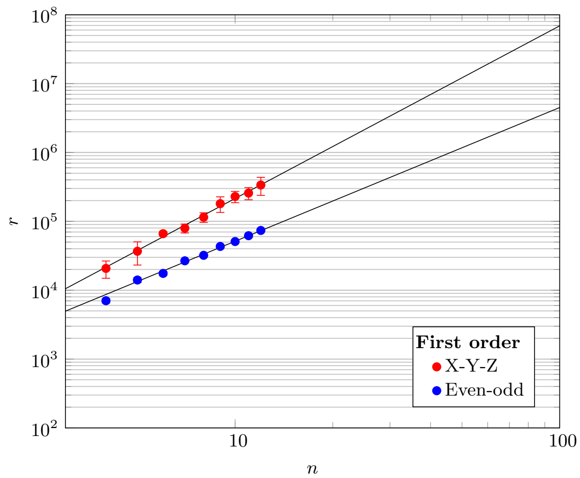

FIG. 2: Comparison of the empirical values of r for the first-order product-formula algorithm with the even-odd ordering and the X-Y-Z ordering of [12]. Error bars are omitted when they are negligibly small on the plot. Straight lines show power-law fits to the data.

[4] W. Auzinger and W. Herfort, Opuscula Mathematica **34**, 243 (2014).

[5] S. Descombes and M. Thalhammer, BIT Numerical Mathematics **50**, 729 (2010).

[6] W. Rossmann, *Lie groups: An introduction through linear groups* (Oxford University Press, 2006).

[7] N. Wiebe, D. Berry, P. Høyer, and B. C. Sanders, Journal of Physics A **43**, 065203 (2010), arXiv:0812.0562.

[8] A. W. Knapp, *Basic real analysis* (Birkhëuser, 2005).

[9] S. Chakraborty, A. Gilyén, and S. Jeffery, "The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation," (2018), arXiv:1804.01973.

[10] R. Bellman, *Introduction to matrix analysis* (Society for Industrial and Applied Mathematics, 1997).

[11] J. Haah, M. B. Hastings, R. Kothari, and G. H. Low, "Quantum algorithm for simulating real time evolution of lattice Hamiltonians," (2018), arXiv:1801.03922.

[12] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Proceedings of the National Academy of Sciences **115**, 9456 (2018), arXiv:1711.10980.