Randomizing multi-product formulas for Hamiltonian simulation

Paul K. Faehrmann[1], Mark Steudtner[1], Richard Kueng[2], Mária Kieferová[3], and Jens Eisert[1,4]

[1]Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany
[2]Institute for Integrated Circuits, Johannes Kepler University Linz, Austria
[3]Centre for Quantum Computation and Communication Technology, Centre for Quantum Software and Information, University of Technology Sydney, NSW 2007, Australia
[4]Helmholtz-Zentrum Berlin für Materialien und Energie, Hahn-Meitner-Platz 1, 14109 Berlin, Germany

Quantum simulation, the simulation of quantum processes on quantum computers, suggests a path forward for the efficient simulation of problems in condensed-matter physics, quantum chemistry, and materials science. While the majority of quantum simulation algorithms are deterministic, a recent surge of ideas has shown that randomization can greatly benefit algorithmic performance. In this work, we introduce a scheme for quantum simulation that unites the advantages of randomized compiling on the one hand and higher-order multi-product formulas, as they are used for example in linear-combination-of-unitaries (LCU) algorithms or quantum error mitigation, on the other hand. In doing so, we propose a framework of randomized sampling that is expected to be useful for programmable quantum simulators and present two new multi-product formula algorithms tailored to it. Our framework reduces the circuit depth by circumventing the need for oblivious amplitude amplification required by the implementation of multi-product formulas using standard LCU methods, rendering it especially useful for early quantum computers used to estimate the dynamics of quantum systems instead of performing full-fledged quantum phase estimation. Our algorithms achieve a simulation error that shrinks exponentially with the circuit depth. To corroborate their functioning, we prove rigorous performance bounds as well as the concentration of the randomized sampling procedure. We demonstrate the functioning of the approach for several physically meaningful examples of Hamiltonians, including fermionic systems and the Sachdev–Ye–Kitaev model, for which the method provides a favorable scaling in the effort.

1 Introduction

The simulation of quantum processes on quantum computers is one of the most eagerly anticipated use cases for quantum computing. The ability to simulate a system's time evolution promises to provide insights into the dynamics of interacting quantum systems in situations where approximate classical simulation methods fail and constitutes one of the cornerstones of quantum technologies [1].

This work aims at improving algorithms that simulate the dynamics of expectation values of observables. The need for developing such machinery stems from the observation that state-of-the-art quantum devices and early quantum computers are still rather limited in their realizable circuit depths and control. We, therefore, assume only access to a quantum-oracle machine that implements single-qubit state preparation, controlled time evolution and quantum measurements and strive for minimizing the required depth of suitable quantum algorithms. Such a setting explicitly allows for the use of product formulas, which are the earliest algorithms proposed for the simulation of time-independent local Hamiltonians [2].

Such Trotter-Suzuki methods, as they are called, have evolved from comparably simple prescriptions for local Hamiltonians to sophisticated schemes able to capture more general sparse time-independent Hamiltonians [3–5] as well as time-dependent Hamiltonians [6, 7] and open quantum systems [8, 9]. Despite their relative simplicity, product formulas are still at the forefront of Hamiltonian simulation

Paul K. Faehrmann: paul.faehrmann@fu-berlin.de
P.K.F. and M.S. have contributed equally.

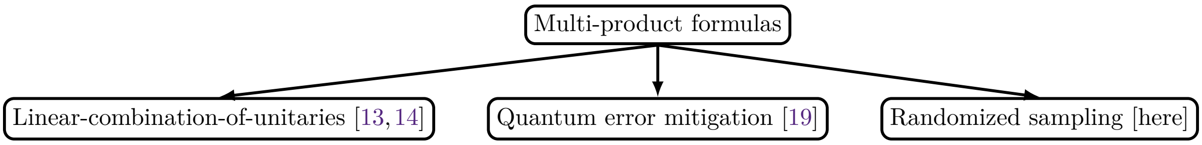

Multi-product formulas
Linear-combination-of-unitaries [13, 14]
Quantum error mitigation [19]
Randomized sampling [here]

Figure 1: Multi-product formulas, although introduced with use in the LCU framework in mind, can also be used in different frameworks, such as quantum error mitigation or randomized sampling, which is the main contribution of this work.

techniques. A numerical study has shown that product formulas can in practice outperform more complex techniques [10] and their complexity is better than initial estimates [11] suggest. In fact, product formulas are nearly optimal for lattice model simulations [12].

Multi-product formulas introduced in the *linear-combinations-of-unitaries (LCU)* [13] approach have been built upon previous results of Trotter-Suzuki methods and have recently been improved [14]. The idea of linearly combining unitaries has led to quantum algorithms with an exponential speedup in precision [15–18] which have been recently gaining popularity. Besides quantum simulation in the LCU framework, the mathematical construct of multi-product formulas (see Fig. 1) can also be used in quantum error mitigation [19]. Although these methods have optimal asymptotic error scaling, they inherently require deep quantum circuits for their implementation and are thus not suitable for applications in near-term devices or early quantum computers. It is therefore crucial to find algorithms that require shorter circuits and fewer digital gates.

Recently, a new element has been introduced to aid this search: the element of *classical randomness*. The idea of randomization in Hamiltonian simulation has heralded a renaissance of product formula methods [20–24]. For single steps of such techniques, a rigorous understanding has recently been reached [24]. Randomized algorithms can also be considered when one is only interested in estimating expectation values. For such a task we do not need to prepare the time-evolved state (from which the observable will be measured) perfectly. Instead, a lower-effort randomized algorithm can be used such that the correct expectation value is obtained only after averaging the measurement outcomes.

The aims of this work are twofold: First, we strive for combining the advantages of higher-order multi-product formulas with those of schemes of randomized compiling, to create a novel framework in which multi-product formulas can be put to good use, which we dub “randomized sampling”. Specifically, we adopt the above scenario of computing expectation values of observables and propose to sample product formulas from multi-product formulas, by which we circumvent the need to use the LCU framework and its corresponding methods such as block encodings. Instead, we implement multi-product formulas on average through random sampling. This results in a notable reduction in algorithmic depth at the expense of additional circuit evaluations. In this way, we see how notions of randomized compiling and higher-order multi-product formulas – when suitably brought together – allow for more resource-efficient notions of quantum simulation amenable to programmable devices and early quantum computers, with a focus on estimating the dynamics of quantum systems instead of a complete phase estimation procedure. An overview of this framework can be found in Figure 2. Second, we develop alternative multi-product formulas tailored to this new framework, which promise to outperform the accuracy of the multi-product formulas introduced by Childs and Wiebe in Ref. [13] in the regime of short simulation times. We show that using a quantum device limited to the aforementioned operations can already yield improvements to fully analog approaches. It is therefore especially useful in the regime of early quantum computers, when algorithms with a pure focus on noisy-intermediate-scale-quantum devices [25], such as variational quantum algorithms, reach their limits, but fully digital algorithms on large, fault-tolerant quantum computers are not yet feasible. The steps required to end up with such a formula are summarized in Figure 3.

A direct comparison between these newly developed multi-product formulas with those of Childs and Wiebe and Trotter-Suzuki product formulas is presented in Table 1.

The remainder of this work is organized as follows: After a short review of multi-product formulas in Section 2, we present the framework of randomized sampling and a summary of our main results in Section 3. Section 4 then contains a detailed analysis of the proposed randomized sampling framework and multi-product formulas and gives the proofs of the results discussed in the previous section. To conclude, we compare the performance of our formulas to that of Trotter-Suzuki product formulas and Childs and Wiebe type multi-product formulas [13] in Section 5, discussing both their error bounds

Algorithm	**Trotter-Suzuki** [26] (Section 2)	**Childs-Wiebe** [13] (Section 2)	**Novel multi-product formulas** [here] (Section 3.2)
Formula	$S_{2\chi}(t/r)^r$	$\sum_{q=1}^{K+1} C_q S_{2\chi}(t/q)^q$	$\prod_{r=1}^{R} \sum_{q=0}^{2\chi R} C_q\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}\right)\ S_{2\chi}\left(b_q^{(r)} t\right)$
Free parameters	$r,\ \chi$	$K,\ \chi$	$R, \chi,\ \{\boldsymbol{b}^{(r)}\}_{r=1}^{R}$
Dependent parameters	-	$\boldsymbol{C}(K, \chi)$	$\left\{\boldsymbol{\nu}^{(r)}(R,\chi),\ \boldsymbol{C}\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}\right)\right\}_{r=1}^{R}$
Max. depth	$2Lr \cdot 5^{\chi-1}$	$2L(K+1) \cdot 5^{\chi-1}$	$2LR \cdot 5^{\chi-1}$
Query complexity per sample	$\mathcal{O}\left(\frac{(\Lambda t)^{1+1/(2\chi)}}{\varepsilon^{1/(2\chi)}} \cdot 5^{\chi-1}\right)$	$\mathcal{O}\left(\frac{\ln(\varepsilon)}{\ln(\Lambda t)} 5^{\chi-1}\right)$	$\mathcal{O}\left(\frac{\ln(\varepsilon)}{\ln(\Lambda t)} \frac{5^{\chi-1}}{\chi}\right)$
Error scaling	$\mathcal{O}\left(r(\Lambda t/r)^{2\chi+1}\right)$	$\mathcal{O}\left((\Lambda t)^{2(\chi+K)+1}\right)$	$\mathcal{O}\left((\Lambda t)^{2\chi R+1}\right)$
Sampling overhead	1	$\left(\sum_{q=1}^{K+1} \lvert C_q\rvert\right)^2$	$\left(\prod_{r=1}^{R} \sum_{q=0}^{2\chi R} \left\lvert C_q\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}\right)\right\rvert\right)^2$

Table 1: Comparison of the standard Trotter-Suzuki formula, the multi-product formula used by Childs and Wiebe and the closed-form/matching multi-product formula introduced in this work for approximating the time evolution operator $\exp(-\mathrm{i}Ht)$ of a Hamiltonian $H = \sum_{k=1}^{L} h_k$. We use $\Lambda = \sum_k \|h_k\|$ and provide detailed error bounds in Section 2 and Theorem 3. Each algorithm provides the choice of some free parameters, dictating their performance and resource requirements. The multi-product formulas proposed by Childs and Wiebe in Ref. [13] and those introduced here further come with additional parameters depending on the choice of these free parameters. Since we aim for a randomized implementation using the framework introduced in Algorithm 1, the maximal circuit depths, query complexities, and error scalings are slightly different from their implementation in a deterministic, coherent fashion. Furthermore, implementing them in the randomized sampling framework comes with the stated sampling overheads. Note that their errors can also exhibit a commutator scaling as discussed in Refs. [11, 14] for all of these methods.

and actual performance on several physically plausible and interesting Hamiltonian models of strongly correlated quantum systems, for which we find a favorable performance over known schemes of quantum simulation.

2 Multi-product formulas

Multi-product formulas constitute the main ingredient of our work, and hence we will briefly review the underlying ideas, starting with product formulas. The goal of quantum simulations is to approximate the quantum dynamics of a complex quantum system described by a many-body Hamiltonian decomposed as

$$H = \sum_{k=1}^{L} h_k \tag{1}$$

composed of Hermitian Hamiltonian terms $\{h_k\}$ of neither necessarily small nor geometrically local support defined on a quantum lattice equipped with a Hilbert space $\mathcal{H} = (\mathbb{C}^d)^{\otimes n}$.

Here, n is the number of degrees of freedom of finite local dimension d. To access the Hamiltonian, we assume to have oracles $\mathbb{O}_k(t)$ implementing time evolutions under each term in the Hamiltonian

$$\mathbb{O}_k(t) \lvert\psi\rangle = e^{-\mathrm{i}h_k t} \lvert\psi\rangle \tag{2}$$

for any $k \in [1, L]$ and $t \in \mathbb{R}$. In digital quantum simulation, these oracles are built from Clifford gates and phase rotations, but we do not assume that such a decomposition is available to us, as the oracles could be implemented by the time evolution of a programmable device. We only require to have control over the implementation, i.e., that we can apply the oracle $\mathbb{O}_k(t)$ depending on the state of a subsystem encoded as a qubit as

$$\lvert 0\rangle\langle 0\rvert \otimes \mathbb{I} + \lvert 1\rangle\langle 1\rvert \otimes \mathbb{O}_k(t)\,. \tag{3}$$

Now, quantum simulation aims at making reliable predictions of the expectation values of observables O (which in the ideal case are local with a small support on the lattice, but again, this is not a necessity) at later times $T > 0$

$$\langle O(T) \rangle := \mathrm{tr}(U(T)\rho U^\dagger(T) O) \tag{4}$$

for initial quantum states ρ, where U is the time evolution operator. In practice, this commonly means to establish ways to approximate the corresponding time evolution operator

$$U(T) = \mathrm{e}^{-\mathrm{i}HT}\,. \tag{5}$$

Although it is worth stressing that a-priori information about the time T and properties of the initial state ρ, as well as features of the locality of the underlying Hamiltonian H can be exploited to come up with highly specialized approximations, we do not assume any underlying structure or limitations in this work.

A simple, yet effective approach to approximating the time evolution operator is that of using product formulas, which make use of N consecutive evolutions of individual Hamiltonian terms h_{k_j} by associated short time intervals $\alpha_j t$, defined with real coefficients α_j such that $\sum_j \alpha_j = 1$. Partial backward evolutions are explicitly allowed, i.e., $\alpha_j < 0$ for some j, even though $t > 0$. In general, such a formula is defined as a product of time evolutions

$$\mathrm{e}^{-\mathrm{i}Ht} \approx \prod_{j=1}^{N} \mathrm{e}^{-\mathrm{i}\alpha_j\, h_{k_j} t} \tag{6}$$

where each single term evolution can be implemented via $\mathbb{O}_k(t)$. Here, we distinguish between a direct evolution by time t and a repeated evolution of short time slices $t/r < 1$ such that

$$\mathrm{e}^{-\mathrm{i}Ht} \approx \left(\prod_{j=1}^{N} \mathrm{e}^{-\mathrm{i}\alpha_j\, h_{k_j} \frac{t}{r}} \right)^r . \tag{7}$$

The most accurate known formulas of this type are *Trotter-Suzuki formulas* [26]. They are recursively defined for any positive integer χ and any time t by

$$S_1(t) = \prod_{k=1}^{L} \mathrm{e}^{-\mathrm{i}h_k t}\,, \tag{8}$$

$$S_2(t) = \prod_{k=1}^{L} \mathrm{e}^{-\mathrm{i}h_k t/2} \prod_{k=L}^{1} \mathrm{e}^{-\mathrm{i}h_k t/2}\,, \tag{9}$$

$$S_{2\chi}(t) = \Big(S_{2\chi-2}(s_{2\chi-2}t)\Big)^2 S_{2\chi-2}\Big([1-4s_{2\chi-2}]t\Big)\Big(S_{2\chi-2}(s_{2\chi-2}t)\Big)^2 \tag{10}$$

with $s_{2\chi} := (4 - 4^{1/(2\chi+1)})^{-1}$ for any positive integer χ. This specific choice of $s_{2\chi}$ ensures that the Taylor series of $S_{2\chi}(t)$ matches that of the actual time evolution $U(t)$ up to $\mathcal{O}(t^{2\chi+1})$, which makes it a good approximation for $t \ll 1$. It is important to stress that constructing an $\mathcal{O}(t^{2\chi+1})$ approximation requires the application of $2 \cdot 5^{\chi-1}(L-1)+1$ individual oracles $\mathbb{O}_k(t)$, where L is again the number of Hamiltonian terms. More generally, we would represent them by oracle calls and we will refer to the number of exponentials N_{exp}. Using repetitions as in Eq. (7), the number of exponentials of the algorithm is of $\mathcal{O}(rL5^{\chi-1})$ to approximate the actual time evolution $U(T)$ up to an accuracy of $\varepsilon \in \mathcal{O}(r(T/r)^{2\chi+1})$ in operator distance.

To enhance the comparability of product formulas with more involved multi-product formulas whose bounds have not been improved to the same extent, we use the following simple tail bound and note that better bounds exist, e.g. via commutator relations [11]:

$$\|U(T) - S_{2\chi}(T/r)^r\| \le 2r \frac{(g_\chi \Lambda T/r)^{2\chi+1}}{(2\chi+1)!} \tag{11}$$

Here, $g_\chi = (5/3)^{\chi-1} 4\chi/3$ originates in the exponential tail of the Trotter-Suzuki terms [13] and $\Lambda = \sum_{k=1}^{L} \|h_k\|$.

The exponential dependence of the circuit depth on χ can pose a challenge for real-world implementations, and especially so within the NISQ regime. Altogether, the number of oracle calls needed to simulate the evolution up to an error $\varepsilon \leq 1$ is upper bounded by

$$N_{\text{oracle}} \leq \frac{2L5^{\chi-1}\left(L\Lambda t\right)^{1+1/(2\chi)}}{\varepsilon^{1/(2\chi)}}, \tag{12}$$

as proven in Ref. [4]. An alternative complexity bound using commutator relations of the individual Hamiltonian terms can be found in Ref. [11].

A known technique to decrease the number of required short time evolutions, i.e., oracle calls, is the use of multi-product formulas [27, 28]. While the Trotter-Suzuki approximation cancels erroneous contributions of higher-order terms by adding backward evolutions, multi-product formulas achieve the same cancellations by superposing different product formulas. Conventionally, one employs multi-product formulas that describe the same time evolution (up to a fixed order of t), but whose erroneous higher-order contributions are of different strength and can thus be made to cancel. This is achieved by approximating the time evolution to the same Trotter-Suzuki order, but considering product formulas different in the number of time slices [27, 29]. The Childs and Wiebe multi-product formula discussed in Ref. [13] is of the form

$$M_{K,2\chi}(t) \coloneqq \sum_{q=1}^{K+1} C_q S_{2\chi}(t/\ell_q)^{\ell_q}, \tag{13}$$

where K is an integer defining a cutoff and $\{\ell_q\}$ is a set of pairwise different integers. The coefficients $\{C_q\}$ are determined via

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \ell_1^{-2\chi} & \ell_2^{-2\chi} & \cdots & \ell_{K+1}^{-2\chi} \\ \ell_1^{-2\chi-2} & \ell_2^{-2\chi-2} & \cdots & \ell_{K+1}^{-2\chi-2} \\ \vdots & \vdots & \ddots & \vdots \\ \ell_1^{-2(K+\chi-1)} & \ell_2^{-2(K+\chi-1)} & \cdots & \ell_{K+1}^{-2(K+\chi-1)} \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ \vdots \\ C_{K+1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{14}$$

ensuring the error terms in the multi-product formula vanish up to $\mathcal{O}(t^{2(K+\chi)})$, resulting in

$$\begin{aligned} \|U(t) - M_{K,2\chi}(t)\| \leq & \left(1 + g_\chi^{2(\chi+K)+1} \sum_{q=1}^{K+1} |C_q|\right) \frac{(\Lambda t)^{2(\chi+K)+1}}{(2(\chi+K)+1)!} \\ =& \mathcal{O}\left((\Lambda t)^{2(K+\chi)+1}\right), \end{aligned} \tag{15}$$

where we have again stated a simple tail bound for comparability and note that it could also exhibit a commutator scaling as discussed in Refs. [11, 14].

Unlike Childs and Wiebe, we will henceforth use the simplest version achieved by setting $\ell_q = q$ for all q since this choice is the most favorable for a randomized implementation, which will become clearer in Section 3.

In Definitions 2 and 3 and Section 4, we employ a different approach for multi-product formulas and develop two techniques whose errors scale with $\mathcal{O}(t^{2\chi R+1})$, and where R is comparable to $K+1$.

Multi-product formulas were firstly used for quantum simulation by Childs and Wiebe [13], who developed the *linear-combinations-of-unitaries (LCU)* approach to directly implement multi-product formulas on a quantum system. Note that sums of unitaries are not inherently physical operations since the unitary group is not closed under addition. Childs and Wiebe use a non-deterministic approach to implement multi-product formulas that can lead to large algorithmic depth. Berry et al. [15] have implemented LCU for a truncated Taylor series nearly perfectly but their use of oblivious amplitude amplification requires an additional register and a complex state preparation procedure. Recently, Ref. [14] has improved the condition number of multi-product formulas and thus extended the use of LCU by amplitude amplification. However, these improvements have no impact on their asymptotic performance and are less favorable for randomized implementations.

Using *randomized sampling* such as proposed in Algorithm 1, we can circumvent the need for these potentially deep circuits. Randomized compiling of multi-product formulas entails sampling among the individual terms of Eq. (13) requiring the implementation of just twice the number of oracle calls, rather than the entire linear combination and the overhead for its implementation.

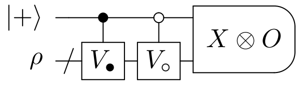
Step 1: Prepare multi-product formula for importance sampling:

$$e^{-iHt} \approx \sum_{k=1}^{M} |C_k| \left(\mathrm{sign}(C_k)\widetilde{V}_k\right) \xrightarrow[\text{divide by } \Xi:=\sum |C_k|]{\text{ensure probability ensemble}} V = \sum_{k=1}^{M} \underbrace{\frac{|C_k|}{\Xi}}_{p_k} \underbrace{(\mathrm{sign}(C_k)\tilde{V}_k)}_{V_k} \approx \frac{1}{\Xi} e^{-iHt}$$

Step 2: Repeat the following two steps N times (see Algorithm 1):

(a) Sample $V_\circ, V_\bullet \overset{i.i.d.}{\sim} \{p_k, V_k\}$ via importance sampling, i.e., draw V_k with probability p_k.

(b) Run the following circuit and measure outcome o_j, with $\mathbb{E}(o_j) = \frac{1}{2}\mathrm{Tr}\left(O(V_\circ \rho V_\bullet^\dagger + V_\bullet \rho V_\circ^\dagger)\right)$:



Main Result: For $N \geq \frac{2\|O\|^2 \ln(2/\delta)}{\epsilon^2}$, we find that $\left|\frac{1}{N}\sum_{j=1}^{N} o_j - \mathrm{tr}(OV\rho V^\dagger)\right| \leq \epsilon$ with confidence δ.

Figure 2: Overview of the randomized sampling framework for estimating the dynamics of observables. After preparing a multi-product formula for importance sampling, we can run Algorithm 1 to implement it in a simple, randomized fashion. While this overview already hints toward its use for time evolution and estimating the dynamics of observables, these results apply to general multi-product formulas.

3 Randomized sampling using multi-product formulas

LCU-type algorithms, despite having a finite failure probability, are deterministic when successfully applied. In contrast to such deterministic implementations, there is a recent interest in randomized algorithms [21–24]. These novel results improve the performance of product formulas by randomizing the order of the short time evolution and introducing the alternative setting of randomized compiling. With the goal of reducing the circuit depth required for the implementation of multi-product formulas, we build upon randomized compiling and propose the setting of randomized sampling: Given an observable O and a quantum system in the initial state ρ, our goal is to find the expectation value of the observable after a time evolution $U = \mathrm{e}^{-\mathrm{i}Ht}$ governed by the Hamiltonian $H = \sum_{k=1}^{L} h_k$. That is, we wish to compute

$$\langle O(t) \rangle = \mathrm{tr}(OU\rho U^\dagger)\,. \tag{16}$$

In the following, we describe a randomized algorithm for such a task, give convergence guarantees, and present novel multi-product formulas suited for this framework. Note that we will require one part of our system to be encoded as a single qubit, while the rest acts as the simulator.

3.1 The randomized sampling framework

We begin by proposing the randomized sampling framework summarized in Figure 2. After rescaling the coefficients of a given multi-product formula, we can apply the following simple algorithm to estimate Eq. (16) in a randomized fashion based upon importance sampling:

Algorithm 1 (Randomized sampling)**.** Given an observable O and an ensemble $\mathcal{V} = \{\,(V_k, p_k)\,\}_{k=1}^{M}$ of M unitaries $\{V_k\}$ and corresponding probabilities $\{p_k\}$, we consider N independent repetitions of the following protocol:

1. Prepare $|+\rangle\!\langle+| \otimes \rho$.

2. Sample $V_\circ \overset{i.i.d.}{\sim} \mathcal{V}$ and apply the anti-controlled unitary $|0\rangle\!\langle 0| \otimes V_\circ + |1\rangle\!\langle 1| \otimes \mathbb{I}$.

3. Sample $V_\bullet \overset{i.i.d.}{\sim} \mathcal{V}$ and apply the controlled unitary $|0\rangle\!\langle 0| \otimes \mathbb{I} + |1\rangle\!\langle 1| \otimes V_\bullet$.

4. Perform a single shot of the POVM measurement associated with the observable $X \otimes O$.

5. Store the measurement outcomes $\{o_j\}$.

Note that it is in general sufficient to measure X and O separately and multiply the outcomes, rather than performing a joint measurement. This could be useful e.g. when the measurement of O is available in a black-box fashion. As we show in more detail in Section 4, we find the following result for infinitely many runs of the algorithm:

Corollary 1 (Sample mean convergence)**.** *Let O be an observable, ρ be an initial state and let $\mathcal{V} = \{ (V_k, p_k) \}_{k=1}^{M}$ be an ensemble of unitaries $\{ V_i \}$ and corresponding probabilities $\{ p_i \}$, such that*

$$\mathbb{E}_{\mathcal{V}}[V_k] = \sum_{k=1}^{M} p_k V_k = V\,. \tag{17}$$

Then, the sample mean of Algorithm 1 converges to the expectation value of the random measurement outcomes $\{o_j\}_j$, given by

$$\lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} o_j = \operatorname{tr}\left(OV\rho V^{\dagger}\right)\,. \tag{18}$$

Furthermore, since an infinite number of measurements is infeasible, we give an expression for the number N of repetitions of Algorithm 1 sufficient for achieving the desired accuracy and confidence for the goal of randomized sampling. Using Hoeffding's inequality [30], we can formulate the following result:

Theorem 1 (Randomized implementation of sums of unitaries)**.** *Let O be an observable, ρ an initial state and let $\mathcal{V} = \{ (V_k, p_k) \}_{k=1}^{M}$ be an ensemble of unitaries $\{ V_k \}$ and corresponding probabilities $\{ p_k \}$, such that*

$$\mathbb{E}_{\mathcal{V}}[V_k] = \sum_{k=1}^{M} p_k V_k = V\,. \tag{19}$$

Then, for a fixed accuracy $\varepsilon \in (0,1)$ and confidence $\delta \in (0,1)$, a total of

$$N \geq \frac{2\|O\|^2 \ln(2/\delta)}{\epsilon^2} \tag{20}$$

repetitions of Algorithm 1 suffice to accurately approximate the expectation value $\operatorname{tr}\left(OV\rho V^{\dagger}\right)$ using the sampling mean estimator, i.e.,

$$\left| \frac{1}{N} \sum_{j=1}^{N} o_j - \operatorname{tr}\left(OV\rho V^{\dagger}\right) \right| \leq \varepsilon \tag{21}$$

with probability at least $1-\delta$.

So far, this framework is fairly general: Given a set $\mathcal{V}$, the algorithm samples from the linear transform with the unitary V. For quantum simulation, V (a decent approximation to U) and $\mathcal{V}$ still have to be found. This is where multi-product formulas come into play. Multi-product formulas approximate U by a weighted sum of product formulas, and so the featured product formulas and their weights could make up the set $\mathcal{V}$. The problem is that we require $\{p_k\}$ to form a probability distribution, i. e., $p_k > 0$ and $\sum_{k=1}^{M} p_k = 1$. This disqualifies us from identifying the sets $\{p_k\}$ with $\{C_q\}$ straightforwardly. While $\sum_q C_q = 1$ holds for any multi-product formula per construction (see the first component of Eq. (14)), the sign of some C_q will be negative [31]. We thus have to absorb the signs of these negative C_q into their unitaries V_q, but then we find that the operation is not normalized as $\sum_q |C_q| > 1$. At this point we introduce the quantity $\Xi := \sum_q |C_q|$, that we call *resolution factor*. The resolution is practically used to define a probability distribution via $p_k = |C_k| / \Xi$ such that Algorithm 1 samples from

$$\frac{1}{\Xi^2} \operatorname{tr}(OU\rho U^{\dagger})\,. \tag{22}$$

As the time evolution is now approximated by ΞV rather than V, Ξ factors into the number of required circuit evaluations. With $\Xi > 1$ requiring to run the algorithm more often to achieve an error comparable with the situation where $\Xi = 1$, this resolution factor can be regarded as a penalty. This penalty is further amplified if a direct time evolution by time t is insufficient and a repeated time evolution of short time slices $t/r < 1$ is required. Since each of such repetitions will result in additional factors of Ξ leading to an exponential scaling of the resolution factor in the number of repetitions, this behavior prohibits the use of a large number of repetitions as is usual for product formulas. On the other hand, this randomized implementation does not rely on post-processing and the increase of some success probability via well-conditioning as is the case for the deterministic, block-encoding implementation, which in turn would increase the number of required circuit evaluations drastically. However, especially in the regime of early quantum computers with capabilities between those of NISQ devices and large fault-tolerant quantum computers, where one would use few repetitions, a deeper circuit may pose a bottleneck much tighter than a slightly higher number of circuit evaluations would. We, therefore, envision these results to be of particular interest in this intermediate regime, where one might be interested in simulating the dynamics of quantum systems instead of performing full-fledged quantum phase estimation.

We can make the following guarantees for approximating time evolutions in randomized sampling with a nontrivial resolution and a finite number of measurements:

Theorem 2 (Approximating unitaries with a finite number of measurements,)**.** *Let U be a unitary, O an observable, ρ some initial state and let $\mathcal{V} = \{ (V_k, p_k) \}_{k=1}^{M}$ be an ensemble of unitaries $\{ V_k \}$ with a corresponding probability distribution $\{ p_k \}$ such that $\mathbb{E}_{\mathcal{V}}[V_k] = \sum_{k=1}^{M} p_k V_k = V$. Let there be a constant factor $\Xi \in \mathbb{R}_+$ such that*

$$\|\Xi V - U\| \leq \varepsilon\,. \tag{23}$$

Then, it is sufficient to run

$$N \geq \frac{2\|O\|^2 \ln(2/\delta)\, \Xi^4}{\varepsilon^2} \tag{24}$$

repetitions of Algorithm 1 to achieve

$$\left| \frac{\Xi^2}{N} \sum_{j=1}^{N} o_j - \mathrm{tr}(O U \rho U^\dagger) \right| \leq (1 + 3\|O\|)\,\varepsilon\,, \tag{25}$$

with probability at least $1 - \delta$.

3.2 Custom multi-product formulas

The multi-product formulas of Childs and Wiebe can easily be adapted to this randomized sampling scheme. Since previous improvements of multi-product formulas focused on increasing success probabilities by changing their condition number [14], which has no impact on their asymptotic error scaling, we can ignore constraints encountered in LCU techniques. Our sole interest lies in a low algorithmic depth and a moderate resolution, so we choose to set $\ell_q = q$ for all $q = 1, \ldots, K+1$ in Eq. (13) and (14). However, those multi-product formulas make relatively little use of the Trotter-Suzuki order 2χ of its components. In fact, the only reason not to minimize the depth by using $S_2(\cdot)$ blocks is the high resolution factor. To this end, we present a family of novel multi-product formulas tailored toward the randomized sampling framework with improved scaling in the Trotter-Suzuki order that can be optimized for their resolution factor. A brief overview of the process leading to a suitable multi-product formula is presented in Figure 3

While Childs and Wiebe's approach manipulates higher-order error terms, we employ a construction to modulate the entire Taylor expansion.

Definition 1 (Linear combination of time evolution operators)**.** Let $\chi \geq 1$ and $R \geq 1$ be integers and $S_{2\chi}(t)$ a Trotter-Suzuki product formula approximation to $U(t)$ as defined in Eq. (10) with $\|S_{2\chi}(t) - U(t)\| \in \mathcal{O}(t^{2\chi+1})$. Then, for any $t \in \mathbb{R}$ and arbitrary $\boldsymbol{b} = (b_0, \ldots, b_{2\chi R})^\top \in \mathbb{R}^{2\chi R+1}$ and $\boldsymbol{\nu} = (\nu_0, \ldots, \nu_{2\chi R})^\top \in \mathbb{R}^{2\chi R+1}$, we define the multi-product formula $\mathcal{L}_{2\chi,R}(\boldsymbol{\nu}, \boldsymbol{b}, t)$ as

$$\mathcal{L}_{2\chi,R}(\boldsymbol{\nu}, \boldsymbol{b}, t) \coloneqq \sum_{q=0}^{2\chi R} C_q(\boldsymbol{\nu}, \boldsymbol{b})\, S_{2\chi}(b_q t)\,, \tag{26}$$

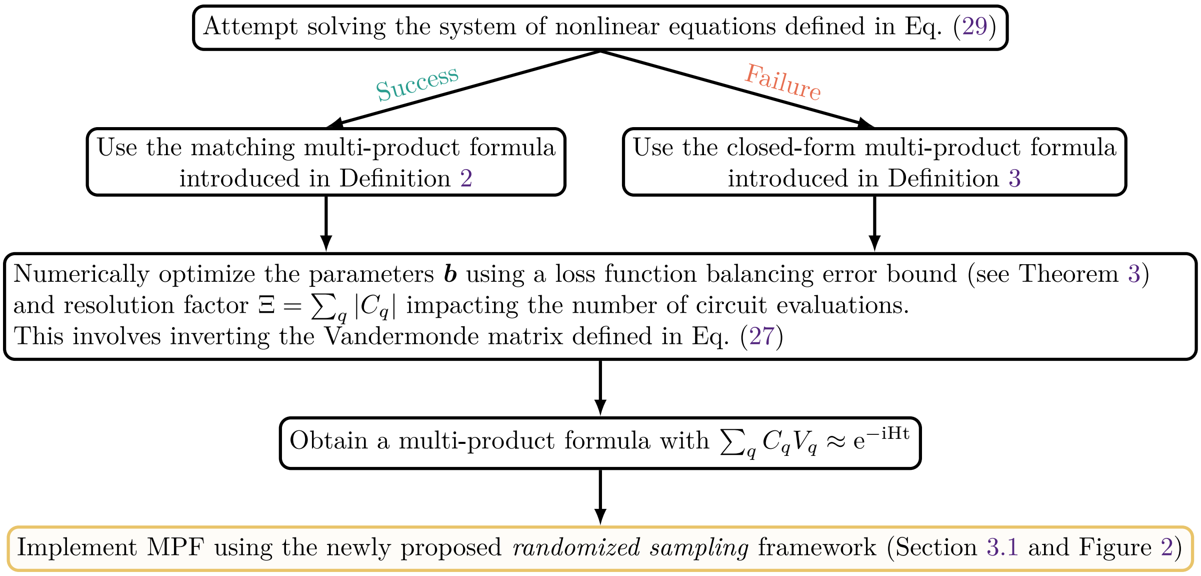

Attempt solving the system of nonlinear equations defined in Eq. (29)
Success
Failure
Use the matching multi-product formula introduced in Definition 2
Use the closed-form multi-product formula introduced in Definition 3
Numerically optimize the parameters $\boldsymbol{b}$ using a loss function balancing error bound (see Theorem 3) and resolution factor $\Xi = \sum_q |C_q|$ impacting the number of circuit evaluations.
This involves inverting the Vandermonde matrix defined in Eq. (27)
Obtain a multi-product formula with $\sum_q C_q V_q \approx \mathrm{e}^{-\mathrm{i}Ht}$
Implement MPF using the newly proposed *randomized sampling* framework (Section 3.1 and Figure 2)

Figure 3: Overview of the steps necessary to arrive at a suitable multi-product formula as discussed in the following section. While the matching multi-product formula is superior to the closed-form version, the required solution to this system of nonlinear equations might not always be available

where

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ b_0 & b_1 & \cdots & b_{2\chi R} \\ b_0^2 & b_1^2 & \cdots & b_{2\chi R}^2 \\ b_0^3 & b_1^3 & \cdots & b_{2\chi R}^3 \\ \vdots & \vdots & \ddots & \vdots \\ b_0^{2\chi R} & b_1^{2\chi R} & \cdots & b_{2\chi R}^{2\chi R} \end{pmatrix} \begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_{2\chi R} \end{pmatrix} = \begin{pmatrix} \nu_0 \\ \nu_1 \\ \nu_2 \\ \nu_3 \\ \vdots \\ \nu_{2\chi R} \end{pmatrix} \tag{27}$$

for some $(C_0, C_1, \ldots, C_{2\chi R})^\top \in \mathbb{R}^{2\chi R+1}$.

Here, the coefficients $\{C_q\}$ are related to the parameters $\boldsymbol{b}$ and $\boldsymbol{\nu}$ via the inverse of the corresponding Vandermonde matrix in Eq. (27) as is discussed in detail in Section 4. The parameters $\boldsymbol{\nu}$ will have an important role in the following construction, whereas $\boldsymbol{b}$ can be tuned to strike a balance between the resolution factor Ξ, error bound ε, and condition number. While previous approaches focused on optimizing $\boldsymbol{b}$ for the condition number relevant for the success probability when using block-encodings and amplitude amplification, we instead numerically optimize them for the resolution factor and obtain improved error bounds along the way. Note that a similar construction has recently been used to approximate energy measurements in the fully analog setting [32]. We can now turn toward the definition of the first new multi-product formula.

Definition 2 (Matching multi-product formula)**.** Let $\chi \geq 1$ and $R \geq 1$ be integers and $\mathcal{L}_{2\chi,R}(\boldsymbol{\nu}, \boldsymbol{b}, t)$ as defined in Definition 1. Then, for any $t \in \mathbb{R}$, we define the multi-product formula $\widetilde{M}^{(\mathrm{m})}_{2\chi,R}(t)$ as

$$\widetilde{M}^{(\mathrm{m})}_{2\chi,R}(t) := \prod_{r=1}^{R} \mathcal{L}_{2\chi,R}\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}, t\right) \tag{28}$$

with $\nu_k^{(r)} = 0$ for $k > 2\chi$ and all $r = 1 \ldots R$. The remaining $\left\{\nu_q^{(r)}\right\}$ are fixed by

$$\sum_{k_1+k_2+\ldots+k_R=k} \frac{\nu_{k_1}^{(1)} \nu_{k_2}^{(2)} \cdots \nu_{k_R}^{(R)}}{k_1!\, k_2! \cdots k_R!} = \frac{1}{k!} \tag{29}$$

for all $0 \leq k \leq 2\chi R$, whereas the parameters $\{\boldsymbol{b}^{(r)}\}$ can be chosen arbitrarily.

Multiplying the coefficients C_q of the individual building blocks $\mathcal{L}_{2\chi,R}(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}, t)$ of the matching multi-product formula results in a resolution factor of

$$\Xi^{(\mathrm{m})} := \prod_r \left(\sum_q \left| C_q\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}\right) \right| \right). \tag{30}$$

For the time evolution to work, the set of vectors $\left\{\boldsymbol{\nu}^{(r)}\right\}$ has to be found satisfying the constraint in Eq. (29) for different R and 2χ. It is important to note that obtaining the coefficients $\boldsymbol{\nu}$ for the matching multi-product formula requires solving the system of nonlinear equations defined by Eq. (29), whose complexity scales at least exponentially in χR. In this work, we have used numerical solvers which converge in a few seconds for $\chi R \leq 10$, which we view as reasonably high as they provide an approximation up to $\mathcal{O}(t^{21})$, which we view as suitable for early quantum computers.

When solving the system of nonlinear equations becomes unfeasible, be it due to slow convergence or the desire for significantly better approximations, we can employ a second version of the multi-product formula in which the $\left\{\boldsymbol{\nu}^{(r)}\right\}$ are already determined at the cost of a slightly larger resolution factor Ξ:

Definition 3 (Closed-form multi-product formula)**.** Let $\chi \geq 1$ and $R \geq 1$ be integers and $\mathcal{L}_{2\chi,R}(\boldsymbol{\nu}, \boldsymbol{b}, t)$ as defined in Definition 1. Then for any $t \in \mathbb{R}$, we define the multi-product formula $\widetilde{M}^{(\mathrm{cf})}_{2\chi,R}(t)$ as

$$\widetilde{M}^{(\mathrm{cf})}_{2\chi,R}(t) := \sum_{r=1}^{R} \left(\mathcal{L}_{2\chi,R}\left(\boldsymbol{\nu}^{(0)}, \boldsymbol{b}^{(0)}, t\right) \right)^{r-1} \mathcal{L}_{2\chi,R}\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}, t\right) \tag{31}$$

with

$$\nu_k^{(0)} = \begin{cases} 1, & \text{for } k = 2\chi \\ 0, & \text{else}, \end{cases} \tag{32}$$

$$\nu_k^{(1)} = \begin{cases} 1, & \text{for } k \leq 2\chi \\ 0, & \text{else}, \end{cases} \tag{33}$$

$$\nu_k^{(n)} = \begin{cases} \frac{k!((2\chi)!)^{n-1}}{(2\chi(n-1)+k)!}, & \text{for } 0 < k \leq 2\chi \\ 0, & \text{else}, \end{cases} \tag{34}$$

for all $0 \leq k \leq 2\chi R$ and $1 < n \leq R$. The parameters $\{\boldsymbol{b}^{(r)}\}$ can be chosen arbitrarily.

Multiplying the coefficients of the individual $\mathcal{L}_{2\chi,R}(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}, t)$, results in a resolution factor for the closed-form multi-product formula of

$$\Xi^{(\mathrm{cf})} := \sum_r \left(\sum_q \left| C_q\left(\boldsymbol{\nu}^{(0)}, \boldsymbol{b}^{(0)}\right) \right| \right)^{r-1} \left(\sum_q \left| C_q\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}\right) \right| \right). \tag{35}$$

For these multi-product formulas, we can prove the following two results:

Corollary 2 (Low depth unitary approximation)**.** *Let $\widetilde{M}_{2\chi,R}(t)$ be a multi-product formula constructed according to Definition 2 or 3. We then find that*

$$\left\| U(t) - \widetilde{M}_{2\chi,R}(t) \right\| = \mathcal{O}\left(t^{2\chi R+1}\right). \tag{36}$$

Theorem 3 (Error bound for custom multi-product formulas)**.** *For $\widetilde{M}_{2\chi,R}(t)$ being a multi-product formula constructed according to Definitions 2 or 3, we have*

$$\left\| U(t) - \widetilde{M}_{2\chi,R}(t) \right\| \leq \left(1 + \zeta g_\chi^{2\chi R+1}\right) \frac{(\Lambda t)^{2\chi R+1}}{(2\chi R+1)!}, \tag{37}$$

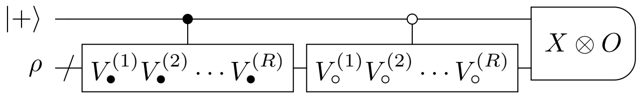

$|+\rangle$
ρ
$V_\bullet^{(1)}V_\bullet^{(2)}\cdots V_\bullet^{(R)}$
$V_\circ^{(1)}V_\circ^{(2)}\cdots V_\circ^{(R)}$
$X\otimes O$

Figure 4: Quantum circuit for randomized sampling with a multi-product formula $\prod_{r=1}^{R}\left(\sum_k p_k^{(r)}V_k^{(r)}\right)$. The unitaries $V_\circ^{(r)}, V_\bullet^{(r)}$ are drawn independently at random from $\{V_k^{(r)}\}_k$ according to the probabilities $\{p_k^{(r)}\}_k$ for all $r = 1\ldots R$, such that the circuit samples from $\mathbb{E}(\langle O\rangle)$. Again, it is in general sufficient to multiply the measurement outcomes instead of performing a joint measurement.

with $g_\chi \ := \frac{4\chi}{3}\left(\frac{5}{3}\right)^{\chi-1}$, $\Lambda := \sum\limits_k \|h_k\|$ *and corresponding*

$$\zeta^{(\mathrm{m})} := \sum_{q_1=0}^{2\chi R}\sum_{q_2=0}^{2\chi R}\cdots\sum_{q_r=0}^{2\chi R}\left|C_{q_1}^{(1)}C_{q_2}^{(2)}\cdots C_{q_r}^{(R)}\right|\left(|b_{q_1}^{(1)}|+\ldots+|b_{q_r}^{(R)}|\right)^{2\chi R+1} \tag{38}$$

$$\leq \Xi^{(\mathrm{m})}\left(R\max_{q,r}|b_q^{(r)}|\right)^{2\chi R} \tag{39}$$

$$\zeta^{(\mathrm{cf})} := \sum_{r=1}^{R}\sum_{q_1=0}^{2\chi R}\sum_{q_2=0}^{2\chi R}\cdots\sum_{q_r=0}^{2\chi R}\left|C_{q_1}^{(0)}C_{q_2}^{(0)}\cdots C_{q_{r-1}}^{(0)}C_{q_r}^{(r)}\right|\left(|b_{q_1}^{(0)}|+\ldots+|b_{q_{r-1}}^{(0)}|+|b_{q_r}^{(r)}|\right)^{2\chi R+1} \tag{40}$$

$$\leq \Xi^{(\mathrm{cf})}\left(R\max_{q,r}|b_q^{(r)}|\right)^{2\chi R}, \tag{41}$$

for matching and closed-form multi-product formulas respectively.

Consequently, these formulas can be used for randomized sampling via Theorem 2. A numerical comparison of the error bounds of all presented formulas can be found in Fig. 5, while an overview of all relevant quantities is provided in Table 1.

While the formula dependent $\zeta_{\chi,R}^{(\mathrm{m})}$ and $\zeta_{\chi,R}^{(\mathrm{cf})}$ themselves are not too illuminating, they can be upper bounded by the corresponding resolution factors multiplied by a term depending on the magnitudes of the used parameters $\boldsymbol{b}$. However, these upper bounds are not tight which is why their exact versions are used in any explicit calculations. Nevertheless, they showcase that a bad choice of $\boldsymbol{b}$ can lead to significant penalties in the performance of the resulting multi-product formula, further stressing the need for optimizing with respect to these parameters.

The matching and closed-form multi-product formulas both rely on products of $\mathcal{L}_{2\chi,R}(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}, t)$. For Algorithm 1, these products can either be expanded and the resulting operators and coefficients be identified with the sets $\{V_k\}$, $\{p_k\}$, or sets of $(V_\circ^{(1)}, V_\bullet^{(1)}), \ldots, (V_\circ^{(R)}, V_\bullet^{(R)})$ be drawn and applied as in Fig. 4. For the matching multi-product formula, all $V_\circ^{(r)}$ and $V_\bullet^{(r)}$ are drawn independently from each other while for the closed-form multi-product formula, the set from which $V_{\circ/\bullet}^{(r+1)}$ is drawn depends on the set from which $V_{\circ/\bullet}^{(r)}$ has been sampled.

Furthermore, since the above framework randomizes only over product formulas, we are not necessarily bound to Trotter-Suzuki formulas and one could also think of a doubly-stochastic version in which the short-term evolutions comprising the product formulas are sampled randomly as well. In that sense, it is allowed to construct the multi-product formulas with building blocks of

$$\widehat{S}_2(t) = \frac{1}{2}\left(S_1(t) + S_1^\dagger(-t)\right), \tag{42}$$

rather than (9). This is not possible for Childs and Wiebe's multi-product formulas, which require symmetric building blocks and are thus limited to Trotter-Suzuki formulas.

4 Proofs

In this section, we provide proofs of the statements presented above. In Section 4.1, we are proving our statements regarding the errors and uncertainty of the randomized sampling procedure. We then turn our attention to the matching and closed-form multi-product formula, providing the intuition behind their construction and verifying their error scaling in Section 4.2. We finally analyze the upper bounds for their error (which is the error of the average time evolution they describe) in Section 4.3.

4.1 Randomized sampling

Let us start by following Algorithm 1 step by step. The controlled and anti-controlled applications of $V_\circ$ and $V_\bullet$ are defined as

$$\overline{\mathrm{C}}V_\circ := |0\rangle\langle 0| \otimes V_\circ + |1\rangle\langle 1| \otimes \mathbb{I}, \tag{43}$$

$$\mathrm{C}V_\bullet := |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes V_\bullet\,, \tag{44}$$

and the initial state ρ will be transformed to

$$\widetilde{\rho} = \mathrm{C}V_\bullet \overline{\mathrm{C}}V_\circ \; (|+\rangle\langle +| \otimes \rho) \; (\overline{\mathrm{C}}V_\circ)^\dagger (\mathrm{C}V_\bullet)^\dagger \tag{45}$$

$$= \frac{1}{2}\left(|0\rangle\langle 0| \otimes V_\circ \, \rho V_\circ^\dagger + |0\rangle\langle 1| \otimes V_\circ \, \rho V_\bullet^\dagger + |1\rangle\langle 0| \otimes V_\bullet \, \rho V_\circ^\dagger + |1\rangle\langle 1| \otimes V_\bullet \, \rho V_\bullet^\dagger\right) \tag{46}$$

after the third step of the protocol. The expectation value for the succeeding measurement of $X \otimes \mathbb{I}$ is then given by

$$\mathrm{tr}\,(X \otimes O\widetilde{\rho}) = \frac{1}{2}\mathrm{tr}\left(O(V_\circ \, \rho V_\bullet^\dagger + V_\bullet \, \rho V_\circ^\dagger)\right). \tag{47}$$

Here, we note that is in general not necessary to perform a joint measurement but sufficient to multiply the measurement results. This might be beneficial for cases in which the measurement of O is only available in a black-box fashion. We now use that $V_\circ$ and $V_\bullet$ are sampled independently from the same ensemble

$$V_\circ, V_\bullet \overset{i.i.d.}{\sim} \mathcal{V} = \{\, (V_k, p_k) \,\}_{k=1}^{M} \tag{48}$$

such that

$$\mathbb{E}_{\mathcal{V}}[V_\circ] = \mathbb{E}_{\mathcal{V}}[V_\bullet] = \sum_{k=1}^{M} p_k V_k = V\,. \tag{49}$$

Then, we can combine the quantum average form Eq. (47) with the classical average from Eq. (49) to conclude that the expectation value of the random single-shot measurement outcome o is given by

$$\mathbb{E}[o] = \mathbb{E}_{\mathcal{V}}\left[\mathrm{tr}\,(X \otimes O\widetilde{\rho})\right] = \mathrm{tr}\left(OV\rho V^\dagger\right), \tag{50}$$

which proves Corollary 1.

For real-world applications, we will never achieve a perfect expectation value. It is therefore vital to inspect the single-shot behavior of Algorithm 1 and give an estimate for the number of iterations required to achieve the desired precision. Since the possible outcomes of measuring X on the auxiliary qubit are ± 1 while the absolute of the outcomes of the O measurement are bounded by $\|O\|$, each single-shot measurement outcome $o_j \in [-\|O\|, \|O\|]$ Consequently, by applying Hoeffding's inequality, we can prove the validity of Theorem 1.

Proof of Theorem 1. Since the single-shot measurement outcomes $\{o_j\}$ obtained from N iterations of Algorithm 1 are individually bounded by the interval $[-\|O\|, \|O\|]$, Hoeffding's inequality states

$$\mathrm{Prob}\left(\left|\frac{1}{N}\sum_{j=1}^{N} o_j - \mathrm{tr}\left(OV\rho V^\dagger\right)\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{N\varepsilon^2}{2\|O\|^2}\right). \tag{51}$$

For a fixed accuracy $\varepsilon \in (0,1)$ and confidence $\delta \in (0,1)$, it is therefore sufficient to perform

$$N \geq \frac{2\|O\|^2 \ln(2/\delta)}{\epsilon^2} \tag{52}$$

repetitions of Algorithm 1 to achieve the desired accuracy and confidence, concluding the proof. □

Given recent developments, for example in Ref. [33], one might wonder whether substantial improvements are possible by using a more refined estimator, most notably median-of-means. Unfortunately, this is not very realistic in most scenarios. For observables that obey $O^2 = \mathbb{I}$, such as local and global Pauli observables, the variance of a single-shot outcome $o_j \in [-1, 1]$ becomes $\mathrm{Var}\left[o_j\right] = 1 - \mathbb{E}\left[o_j\right] = \mathcal{O}(1)$. In this case, the variance is of the same order as the magnitude and it is impossible to (asymptotically) improve over Hoeffding's inequality (asymptotic normality) [34]. However, median-of-means could still be used in this setting to take advantage of additional information about the variance, in cases where it is available. Now, we additionally assume that V times the resolution Ξ approximates the time evolution operator U, i.e.,

$$\|\Xi V - U\| \leq \varepsilon \,. \tag{53}$$

Furthermore, we will use the following result.

Lemma 1 (Closeness of expectation values)**.** *Let U be unitary and V and Ξ as given above such that Eq. (53) holds. Furthermore, fix a state ρ and observable O. Then,*

$$\left|\mathrm{tr}\left(OU\rho U^\dagger\right) - \Xi^2 \mathrm{tr}\left(OV\rho V^\dagger\right)\right| \leq 3\varepsilon\|O\| \,. \tag{54}$$

Proof. We begin by defining $\widetilde{U}$ as the difference between the exact and approximated time evolution

$$\Xi V = \Xi \sum_{k=1}^{M} p_k V_k = U + \widetilde{U}. \tag{55}$$

According to Eq. (53) we find that $\|\widetilde{U}\| \leq \varepsilon$. Consequently,

$$\begin{aligned}\left|\mathrm{tr}\left(OU\rho U^\dagger\right) - \Xi^2 \mathrm{tr}\left(OV\rho V^\dagger\right)\right| &= \left|\mathrm{tr}\left(\widetilde{U}^\dagger O U \rho\right) + \mathrm{tr}\left(U^\dagger O \widetilde{U} \rho\right) + \mathrm{tr}\left(\widetilde{U}^\dagger O \widetilde{U} \rho\right)\right| \\ &\leq 2\|\widetilde{U}\|\|U\|\|O\| + \|\widetilde{U}\|^2\|O\| \qquad (56) \\ &\leq \left(2\varepsilon + \varepsilon^2\right)\|O\| \qquad (57) \\ &\leq 3\varepsilon\|O\|. \qquad (58)\end{aligned}$$

□

By combining the insights from the previous discussion, we can now tackle Theorem 2:

Proof of Theorem 2. Add and subtract $\Xi^2 \mathrm{tr}\left(OV\rho V^\dagger\right)$ to find

$$\left|\frac{\Xi^2}{N}\sum_{j=1}^{N} o_j - \mathrm{tr}\left(OU\rho U^\dagger\right)\right| \leq \Xi^2 \left|\frac{1}{N}\sum_{j=1}^{N} o_j - \mathrm{tr}\left(OV\rho V^\dagger\right)\right| + \left|\Xi^2 \mathrm{tr}\left(OV\rho V^\dagger\right) - \mathrm{tr}\left(OU\rho U^\dagger\right)\right|. \tag{59}$$

By applying Theorem 1 with accuracy ε/Ξ^2 to the first term and using Lemma 1, we find that this upper bound is given by $(3\|O\| + 1)\varepsilon$. □

4.2 Construction of the new multi-product formulas

In Definitions 2 and 3, we propose two alternatives to Childs and Wiebe's multi-product formulas, which reduce the number of circuit evaluations required for a randomized implementation and have improved error scaling. In the following, we will motivate their construction and stress the advantages they provide. First of all, note that every approximation of the exact time evolution $\widetilde{U}(t) \approx \mathrm{e}^{-\mathrm{i}tH}$ can be written as a sum of operators $\hat{A}_k$ such that

$$\widetilde{U}(t) = \sum_{k=0}^{\infty} \hat{A}_k \, t^k \,, \tag{60}$$

with $\hat{A}_0 = \mathbb{I}$. For any approximated time evolution with an error of $\mathcal{O}(t^{m+1})$, we find that $\hat{A}_k = (iH)^k/k!$ for all $k \leq m$. In other words, the Taylor expansion of the exact time evolution and its approximation has the same Taylor expansion for the first m non-trivial terms.

For the standard Trotter-Suzuki formula with $\widetilde{U}(t) = S_{2\chi}(t)$, we have

$$\hat{A}_k = (iH)^k/k! \tag{61}$$

for all $k \leq 2\chi$. For $k > 2\chi$, these operators $\hat{A}_k$ resemble uncontrolled, erroneous operators. In Definition 1, we propose a superposition of approximations with differently scaled times $S_{2\chi}(b_n t)$, introducing controllable modulation parameters ν_k up to a Taylor order of $2\chi R$. Its Taylor expansion is given by

$$\begin{aligned} \mathcal{L}_{2\chi,R}(\boldsymbol{\nu}, \boldsymbol{b}, t) &= \sum_{q=0}^{2\chi R} C_q(\boldsymbol{\nu}, \boldsymbol{b})\, S_{2\chi}(b_q t) &(62)\\ &= \sum_{k=0}^{\infty} \left(\sum_{q=0}^{2\chi R} C_q(\boldsymbol{\nu}, \boldsymbol{b})\, b_q^k \right) \hat{A}_k t^k &(63)\\ &= \sum_{k=0}^{2\chi R} \nu_k\, \hat{A}_k t^k \;+\; \mathcal{O}\left(t^{2\chi R+1}\right), &(64) \end{aligned}$$

where $C_q, b_q \in \mathbb{R}$ and $\nu_k = \left(\sum_q C_q\, b_q^k\right)$ is fixed via the linear transformation of the coefficients

$$\boldsymbol{C} = (C_0, C_1, \ldots, C_{2\chi R})^{\top}$$

with the Vandermonde matrix $B_{j,k} = b_k^{j-1}$, $B\boldsymbol{C} = \boldsymbol{\nu}$ as in Eq. (27). The condition number of the Vandermonde matrix as the product of its Hilbert-Schmidt norm and the norm of the pseudo-inverse can be bounded from above and below by explicit expressions involving the vector defining the Vandermonde matrix [35]. Choosing the vector $\boldsymbol{b}$, we can calculate the coefficients $\boldsymbol{C}$ for a fixed solution vector $\boldsymbol{\nu}$ using the inverse Vandermonde matrix B^{-1}, which is found exactly to be [36, 37]

$$\left(B^{-1}\right)_{j,k} = \frac{(-1)^{k-1}}{\prod\limits_{m\,\in\,\boldsymbol{\mu}(j)} (b_m - b_j)} \sum_{\substack{\boldsymbol{a}\in\mathbb{F}_2^{2\chi R}\\ |\boldsymbol{a}|=2\chi R-k}} \prod_{i=0}^{2\chi R-1} \left(b_{\mu_i(j)}\right)^{a_i} \tag{65}$$

$$\text{with} \quad \boldsymbol{\mu}(j) = (0, 1, \ldots, j-1, j+1, \ldots, 2\chi R)\,,$$

where the sum runs over all binary strings $\boldsymbol{a} = (a_0\, a_1\, \ldots\, a_{2\chi R-1})$ of length $2\chi R$ and Hamming weight $2\chi R - k$. However, we have found that for numerical purposes, a matrix inversion of B clearly outmatches the analytical computation of B^{-1} in terms of runtime.

Using Eq. (64), we can now manipulate the Taylor expansion of a Trotter-Suzuki block $S_{2\chi}(t)$ by replacing it with some $\mathcal{L}_{2\chi,R}(\boldsymbol{\nu}, \boldsymbol{b}, t)$. The key insight here is that while the operators $\hat{A}_k$ are only correct for Taylor orders $k \leq 2\chi$, we gain control of the prefactors up to order $2\chi R$. Consequently, we can eliminate all erroneous $\hat{A}_k$ for $2\chi < k \leq 2\chi R$ by setting the corresponding ν_k to zero. This allows us to construct the matching and closed-form multi-product formulas using blocks of $\mathcal{L}_{2\chi,R}(\boldsymbol{\nu}, \boldsymbol{b}, t)$ with different $\boldsymbol{\nu}$ (and possibly $\boldsymbol{b}$).

4.2.1 Matching multi-product formula

The first version of our proposed multi-product formula builds upon the multiplication of R multi-product building blocks $\mathcal{L}_{2\chi,R}(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}, t)$ that according to Eq. (64) can be written as

$$\mathcal{L}_{2\chi,R}\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}, t\right) = \sum_{k=0}^{2\chi} \frac{\nu_k^{(r)}}{k!} \left(-\mathrm{i}Ht\right)^k + \sum_{k=2\chi+1}^{2\chi R} \nu_k^{(r)} \hat{A}_k t^k + \mathcal{O}\left(t^{2\chi R+1}\right)\,, \tag{66}$$

where we can eliminate the second sum by setting $\nu_k = 0$ for $2\chi < k \leq 2\chi R$. Their product now yields

$$\prod_{r=1}^{R} \mathcal{L}_{2\chi,R}\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}, t\right) = \sum_{k=0}^{2\chi R} \mu_k \left(-\mathrm{i}Ht\right)^k + \mathcal{O}(t^{2\chi R+1}) \tag{67}$$

with

$$\mu_k = \sum_{i_1+i_2+\ldots+i_R=k} \frac{\nu_{i_1}^{(1)} \nu_{i_2}^{(2)} \cdots \nu_{i_R}^{(R)}}{i_1! i_2! \cdots i_R!}\,. \tag{68}$$

To mimic the exact time evolution up to $\mathcal{O}(t^{2\chi R+1})$, we require $\mu_k = 1/k!$, leading to Definition 2.

4.2.2 Closed-form multi-product formula

The closed-form version of the presented multi-product formula sums products of building blocks $\mathcal{L}$. To motivate the specific construction, it is useful to take a look at the specific case of $2\chi = 4$ and $R = 3$ for some choice of $\boldsymbol{b}$ and t using the shorthand

$$\mathcal{L}\begin{pmatrix} \nu_0 \\ \vdots \\ \nu_{2\chi R} \end{pmatrix} := \mathcal{L}(\boldsymbol{\nu}, \boldsymbol{b}, t), \tag{69}$$

to get

$$\widetilde{M}^{(\mathrm{cf})}_{4,3} = \mathcal{L}\begin{pmatrix} 1 \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ 0 \\ 0 \\ \vdots \end{pmatrix} + 4!\,\mathcal{L}\begin{pmatrix} 0 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \\ 0 \\ 0 \\ \vdots \end{pmatrix}\mathcal{L}\begin{pmatrix} 0 \\ \mathbf{1!/5!} \\ \mathbf{2!/6!} \\ \mathbf{3!/7!} \\ \mathbf{4!/8!} \\ 0 \\ 0 \\ \vdots \end{pmatrix} + (4!)^2\mathcal{L}\begin{pmatrix} 0 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \\ 0 \\ 0 \\ \vdots \end{pmatrix}\mathcal{L}\begin{pmatrix} 0 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \\ 0 \\ 0 \\ \vdots \end{pmatrix}\mathcal{L}\begin{pmatrix} 0 \\ \mathbf{1!/9!} \\ \mathbf{2!/10!} \\ \mathbf{3!/11!} \\ \mathbf{4!/12!} \\ 0 \\ 0 \\ \vdots \end{pmatrix}, \tag{70}$$

where the orders $1, \dots, 2\chi$ have been visually highlighted for clarity. In this example, the eleventh order in t is revealed by multiplying the corresponding terms from the Taylor expansion of $\mathcal{L}$ in Eq. (62), with the corresponding coefficients in (70)

$$(4!)^2 \times \frac{(-\mathrm{i}Ht)^4}{4!} \times \frac{(-\mathrm{i}Ht)^4}{4!} \times 3!/11! \times \frac{(-\mathrm{i}Ht)^3}{3!} = \frac{(-\mathrm{i}Ht)^{11}}{11!}. \tag{71}$$

The first term of Eq. (70) takes care of the zeroth and the first 2χ order of U. The second term multiplies all terms of orders $t^1 \dots t^{2\chi}$ with $t^{2\chi}$ and so takes care of the next 2χ terms of the expansion. The third term multiplies with a $\mathcal{O}(t^{2\chi})$ term twice, taking care of the subsequent 2χ terms – a pattern emerges. In a general setting (with arbitrary 2χ and R) we can write this sum as

$$\begin{aligned} \widetilde{M}^{(\mathrm{cf})}_{2\chi,R}(t) &:= \sum_{r=1}^{R} \left(\mathcal{L}_{2\chi,R}\left(\boldsymbol{\nu}^{(0)}, \boldsymbol{b}^{(0)}, t\right)\right)^{r-1} \mathcal{L}_{2\chi,R}\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}, t\right) \\ &= \sum_{r=1}^{R} \left(\sum_{j=0}^{2\chi R} \nu_j^{(0)} \frac{(-\mathrm{i}Ht)^j}{j!} + \mathcal{O}\left(t^{2\chi R+1}\right)\right)^{r-1} \left(\sum_{k=0}^{2\chi R} \nu_k^{(r)} \frac{(-\mathrm{i}Ht)^k}{k!} + \mathcal{O}\left(t^{2\chi R+1}\right)\right) && (72) \\ &= \sum_{r=1}^{R} \left(\frac{(-\mathrm{i}Ht)^{2\chi}}{(2\chi)!}\right)^{r-1} \sum_{k=0}^{2\chi R} \nu_k^{(r)} \frac{(-\mathrm{i}Ht)^k}{k!} + \mathcal{O}\left(t^{2\chi R+1}\right) && (73) \end{aligned}$$

where we have used Eq. (64) in Eq. (72) and $\nu_j^{(0)} = \delta_{2\chi,j}$ from Definition 3 to collapse the first factor into $(-\mathrm{i}Ht)^{2\chi(r-1)}/(2\chi)!$ in Eq. (73). Considering also that $\nu_j^{(0)} = 1$ for $0 \leq j \leq 2\chi$, the $r = 1$ term (that we recognize are the first 2χ orders of the time evolution) can be separated from the sum. Discarding all terms that vanish due to $\nu_k^{(r)} = 0$ we rewrite Eq. (73) to

$$\widetilde{M}^{(\mathrm{cf})}_{2\chi,R}(t) = \sum_{j=0}^{2\chi} \frac{(-\mathrm{i}Ht)^j}{j!} + \sum_{r=2}^{R}\sum_{k=1}^{2\chi} \nu_k^{(r)} \frac{(-\mathrm{i}Ht)^{2\chi(r-1)+k}}{((2\chi)!)^{r-1}\,k!} + \mathcal{O}\left(t^{2\chi R+1}\right), \tag{74}$$

for which we consult Definition 3, a last time resolving the remaining $\nu_k^{(r)}$. This leaves us with the correct time evolution up to order $2\chi R + 1$, thus proving the definition.

4.3 Error of the averaged operators

Following upon the insights from Section 4.2 we find Corollary 2 already proven by Eq. (66) and Eq. (74) as long as $\widetilde{M}_{2\chi,R}$ is constructed according to Definition 2 or 3. The proof of Theorem 3 requires a more involved error analysis.

Proof of Theorem 3. We first need to bound the remainder terms of the Taylor series expansions of $U(t)$ and $\widetilde{M}^{(\mathrm{m})}$. In the following, $\mathbf{R}_\ell(f)$ denotes the remainder term of the Taylor series of an operator-valued function f truncated at ℓ-th order in t. We thus find that

$$\begin{aligned} \left\| U(t) - \widetilde{M}^{(\mathrm{m})}_{2\chi,R} \right\| &= \left\| \mathrm{e}^{-\mathrm{i}tH} - \prod_{r=1}^{R} \left[\sum_{q=0}^{2\chi R} C_q\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}\right) \, S_{2\chi}\left(b_q^{(r)} t\right) \right] \right\| \\ &\leq \left\| \mathbf{R}_{2\chi R}\left(\mathrm{e}^{-\mathrm{i}tH}\right) \right\| + \left\| \mathbf{R}_{2\chi R}\left(\prod_{r=1}^{R} \left[\sum_{q=0}^{2\chi R} C_q\left(\boldsymbol{\nu}^{(r)}, \boldsymbol{b}^{(r)}\right) \, S_{2\chi}\left(b_q^{(r)} t\right) \right] \right) \right\| . \end{aligned} \tag{75}$$

Moving forward, we employ some recently established results on the theory of Trotter errors. Specifically, we make use of the 'Trotter error with 1-norm scaling' lemma of Ref. [11]. Building upon these insights in this fresh context, we find that the exponential remainder of a product formula such as in Eq. (6) can be bounded by

$$\begin{aligned} \left\| \mathbf{R}_\ell \left(\prod_{j=1}^{N} \mathrm{e}^{-\mathrm{i}\alpha_j h_{k_j} t} \right) \right\| &= \frac{t^{\ell+1}}{(\ell+1)!} \left\| \left(\frac{\partial}{\partial t} \right)^{\ell+1} \prod_{j=1}^{N} \mathrm{e}^{-\mathrm{i}\alpha_j h_{k_j} t} \right\| \\ &= \frac{t^{\ell+1}}{(\ell+1)!} \left\| \sum_{x_1+\cdots+x_N=\ell+1} \frac{(\ell+1)!}{x_1! \cdots x_N!} \prod_{j=1}^{N} \left(\frac{\partial}{\partial t} \right)^{x_j} \mathrm{e}^{-\mathrm{i}\alpha_j h_{k_j} t} \right\| \\ &= \frac{t^{\ell+1}}{(\ell+1)!} \left\| \sum_{x_1+\cdots+x_N=\ell+1} \frac{(\ell+1)!}{x_1! \cdots x_N!} \prod_{j=1}^{N} \left(-\mathrm{i}\alpha_j \, h_{k_j} \right)^{x_j} \mathrm{e}^{-\mathrm{i}\alpha_j h_{k_j} t} \right\| \\ &\leq \frac{t^{\ell+1}}{(\ell+1)!} \sum_{x_1+\cdots+x_N=\ell+1} \frac{(\ell+1)!}{x_1! \cdots x_N!} \prod_{j=1}^{N} \left\| \alpha_j \, h_{k_j} \right\|^{x_j} \cdot \underbrace{\left\| \mathrm{e}^{-\mathrm{i}\alpha_j h_{k_j} t} \right\|}_{=1} \\ &= \frac{\left(\sum_{j=1}^{N} \left\| \alpha_j \, h_{k_j} \right\| t \right)^{\ell+1}}{(\ell+1)!} . \end{aligned} \tag{76}$$

This bound can now be applied to (75). For the first term, we obtain

$$\left\| \mathbf{R}_{2\chi R}\left(\mathrm{e}^{-\mathrm{i}tH}\right) \right\| \leq \frac{(\Lambda t)^{2\chi R+1}}{(2\chi R+1)!}, \tag{77}$$

while the second term can be bounded by

$$\begin{aligned} \left\| \mathbf{R}_{2\chi R} \left(\prod_{r=1}^{R} \left[\sum_{q=0}^{2\chi R} C_q^{(r)} S_{2\chi}\left(b_q^{(r)} t\right) \right] \right) \right\| &\leq \sum_{q_1,\ldots,q_R=0}^{2\chi R} \left| C_{q_1}^{(1)} \cdots C_{q_R}^{(R)} \right| \left\| \mathbf{R}_{2\chi R}\left(S_{2\chi}\left(b_{q_1}^{(1)} t\right) \cdots S_{2\chi}\left(b_{q_R}^{(R)} t\right) \right) \right\| \\ &\leq \frac{(g_\chi \Lambda t)^{2\chi R+1}}{(2\chi R+1)!} \sum_{q_1,\ldots,q_R=0}^{2\chi R} \left| C_{q_1}^{(1)} \cdots C_{q_R}^{(R)} \right| \left(\sum_{i=1}^{R} \left| b_{q_i}^{(i)} \right| \right)^{2\chi R+1} \end{aligned} \tag{78}$$

where we have introduced the factor

$$g_\chi := \frac{4\chi}{3} \left(\frac{5}{3} \right)^{\chi-1} \tag{79}$$

to group all terms relating to χ from the Trotter-Suzuki decomposition appearing in Eq. (53) of Ref. [13].

At the same time, we have defined $\Lambda := \sum_k \|h_k\|$. Consequently, we can bound the error of the matching multi-product formula via

$$\left\| U(t) - \widetilde{M}^{(\mathrm{m})}_{2\chi,R}(t) \right\| \leq \left(1 + \zeta^{(\mathrm{m})}_{\chi,R} g_\chi^{2\chi R+1} \right) \frac{(\Lambda t)^{2\chi R+1}}{(2\chi R+1)!} , \tag{80}$$

with

$$\zeta^{(\mathrm{m})}_{\chi,R} := \sum_{q_1=0}^{2\chi R}\sum_{q_2=0}^{2\chi R}\cdots\sum_{q_R=0}^{2\chi R}\left|C^{(1)}_{q_1}C^{(2)}_{q_2}\cdots C^{(R)}_{q_R}\right|\left(|b^{(1)}_{q_1}|+\ldots+|b^{(R)}_{q_R}|\right)^{2\chi R+1} \tag{81}$$

The error analysis of the closed-form multi-product formula follows along similar lines: Again, we bound the remainder terms of the Taylor series expansions of $U(t)$ and $\widetilde{M}^{(\mathrm{cf})}$, finding

$$\begin{aligned}
&\left\|U(t)-\widetilde{M}^{(\mathrm{cf})}_{2\chi,R}\right\| = \left\|\mathrm{e}^{-\mathrm{i}tH}-\sum_{r=1}^{R}\left(\mathcal{L}_{2\chi,R}\left(\boldsymbol{\nu}^{(0)},\boldsymbol{b}^{(0)},t\right)\right)^{r-1}\mathcal{L}_{2\chi,R}\left(\boldsymbol{\nu}^{(r)},\boldsymbol{b}^{(r)},t\right)\right\|\\
&\leq \left\|\mathbf{R}_{2\chi R}\left(\mathrm{e}^{-\mathrm{i}tH}\right)\right\|\\
&\quad+\left\|\mathbf{R}_{2\chi R}\left(\sum_{r=1}^{R}\left[\sum_{q=0}^{2\chi R}C_q\left(\boldsymbol{\nu}^{(0)},\boldsymbol{b}^{(0)}\right)\,S_{2\chi}\left(b^{(0)}_q t\right)\right]^{r-1}\left[\sum_{q=0}^{2\chi R}C_q\left(\boldsymbol{\nu}^{(r)},\boldsymbol{b}^{(r)}\right)\,S_{2\chi}\left(b^{(r)}_q t\right)\right]\right)\right\|,
\end{aligned} \tag{82}$$

where the second term can be bounded by

$$\begin{aligned}
&\left\|\mathbf{R}_{2\chi R}\left(\sum_{r=1}^{R}\left[\sum_{q=0}^{2\chi R}C_q\left(\boldsymbol{\nu}^{(0)},\boldsymbol{b}^{(0)}\right)\,S_{2\chi}\left(b^{(0)}_q t\right)\right]^{r-1}\left[\sum_{q=0}^{2\chi R}C_q\left(\boldsymbol{\nu}^{(r)},\boldsymbol{b}^{(r)}\right)\,S_{2\chi}\left(b^{(r)}_q t\right)\right]\right)\right\|\\
&\leq \sum_{r=1}^{R}\sum_{q_1=0}^{2\chi R}\cdots\sum_{q_r=0}^{2\chi R}\left|C_{q_1}(\boldsymbol{\nu}^{(0)},\boldsymbol{b}^{(0)})\cdots C_{q_{r-1}}(\boldsymbol{\nu}^{(0)},\boldsymbol{b}^{(0)})\,C_{q_r}(\boldsymbol{\nu}^{(r)},\boldsymbol{b}^{(r)})\right|\\
&\qquad\qquad\times\left\|\mathbf{R}_{2\chi R}\left(S_{2\chi}\left(b^{(0)}_{q_1}\right)\cdots S_{2\chi}\left(b^{(0)}_{q_{r-1}}\right)\,S_{2\chi}\left(b^{(r)}_{q_r}\right)\right)\right\|
\end{aligned} \tag{83}$$

Defining

$$\begin{aligned}
\zeta^{(\mathrm{cf})}_{\chi,R} &:= \sum_{r=1}^{R}\sum_{q_1=0}^{2\chi R}\sum_{q_2=0}^{2\chi R}\cdots\sum_{q_r=0}^{2\chi R}\left|C^{(0)}_{q_1}C^{(0)}_{q_2}\cdots C^{(0)}_{q_{r-1}}C^{(r)}_{q_r}\right|\left(|b^{(0)}_{q_1}|+\ldots+|b^{(0)}_{q_{r-1}}|+|b^{(r)}_{q_r}|\right)^{2\chi R+1}, &(84)\\
\Lambda &:= \sum_k\|h_k\|, &(85)
\end{aligned}$$

we find

$$\left\|U(t)-\widetilde{M}^{(\mathrm{cf})}_{2\chi,R}(t)\right\| \leq \left(1+\zeta^{(\mathrm{cf})}_{\chi,R}g_\chi^{2\chi R+1}\right)\frac{(\Lambda t)^{2\chi R+1}}{(2\chi R+1)!}, \tag{86}$$

concluding the proof. □

5 Comparison and numerical validation

We will now compare the Trotter-Suzuki product formula algorithm and multi-product formulas in the randomized sampling framework. To make comparisons as fair as possible, we will assume that each algorithm uses at most R-fold sequences of $S_{2\chi}(\cdot)$ blocks. Product formulas may use repetitions as in Eq. (7) achieving an error of $\mathcal{O}((t/R)^{2\chi+1})$, which is a reliable way to approximate time evolutions for longer times $R \gg \tau > 1$. However, repetitions improve the accuracy of shorter time evolutions only minimally. The situation is different if R is an integer power of five, which allows us to build the next higher Trotter-Suzuki order and approximate the time evolution up to a leading order of $2(\chi+\log_5 R)+1$ in t. This means improving the leading power of t comes at an exponential cost for the circuit depth. The situation can be remedied by sampling from a multi-product formula. Remarkably, the statement for sampling observable eigenvalues with product formulas is very similar to Theorem 2.

Since the sampling errors of all of these methods are comparable, we disregard them in this comparison and instead only think about asymptotic limits. The number of required circuit evaluations for the same accuracy then follows directly from the corresponding resolution factors. The main difference between product and multi-product formulas is that Trotter-Suzuki algorithms do not have a resolution

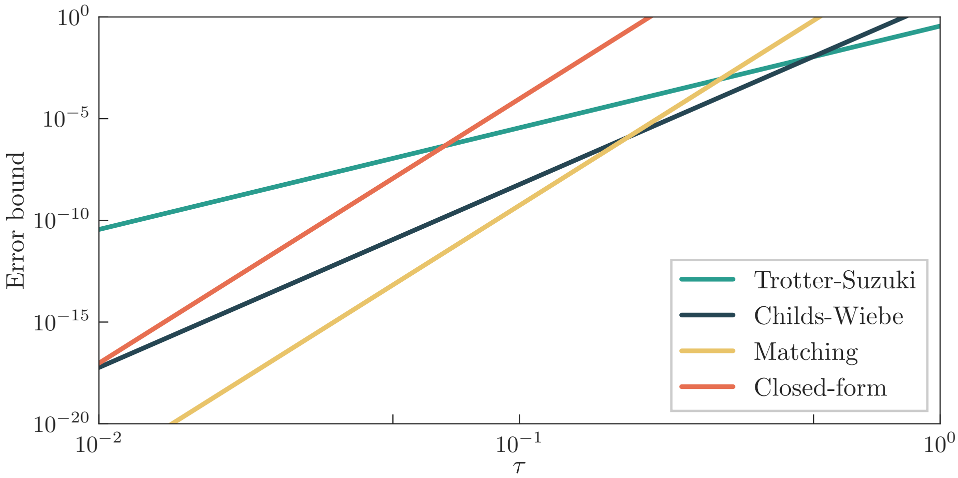


Figure 5: Comparing error bounds for Trotter-Suzuki product formulas as well as several multi-product formulas for different $\tau = \Lambda t$. The exact formulas for these bounds are provided in Equations (11) and (15) and Theorem 3. We have chosen $2\chi = 4$ and $r = R = K + 1 = 3$ to fix the depth of all methods to be 30 oracle calls. The parameters $\{\boldsymbol{b}^{(r)}\}$ have been numerically optimized for the matching and closed-form multi-product formulas with an initial guess of $\boldsymbol{b} = \{1, -1, 2, -2, \ldots, 7\}$, while we employ $\ell_q = q$ for Childs and Wiebe's formula as the choice with the lowest resolution factor. We find the resolution factors for the multi-product formulas to be $\Xi^{(\mathrm{CW})} \approx 3.13$, $\Xi^{(\mathrm{m})} \approx 1.22$ and $\Xi^{(\mathrm{cf})} \approx 1.36$. Consequently, the matching and closed-form multi-product formulas would allow for four to five repetitions before they require the same amount of circuit evaluations as Childs and Wiebe's formula. In the black box optimization, we have used the error bound as the objective function with the modification of using $(\Xi^{(\mathrm{m})})^{20}$ and $(\Xi^{(\mathrm{cf})})^{10}$, instead of $\zeta_{\chi,R}$ respectively, to ensure reasonable resolution factors.

factor. This is equivalent to $\Xi = 1$ with consequences for sampling complexity and error bounds. This advantage is, however, quickly outweighed as Childs and Wiebe's multi-product formula delivers an improved approximation that is exact up to a leading order of $2(\chi + R) - 1$ in t. Modifying the leading power of t is now possible by adding exponentially fewer terms. Here, we consider only the simplest form of Childs and Wiebe's multi-product formula with $\ell_q = q$ since their improvements, e.g. those proposed in Ref. [14], focus on increasing the success probability of their deterministic implementation without improving their error scaling and lead to significantly worse resolution factors. Furthermore, note that they require the same circuit depth for $R = K + 1$ in the randomized sampling framework due to the final term in Eq. (13). With matching and closed-form multi-product formulas, the approximation can be further improved to leading order $2\chi R + 1$ in t. This means that only one additional $S_{2\chi}(\cdot)$ block improves the order by 2χ rather than 2 as for multi-product formulas of the prior art. While the scaling in t of the novel multi-product formulas is superior to those of Childs and Wiebe, their theoretical error bound is not as tight, leading to a crossover of error bounds at $\sum |h_k| t < 1$. While it is true that the error bounds cross over at comparably small errors, it is important to note that the significantly reduced resolution factors allow for multiple repetitions, i.e., r evolutions of shorter times t/r, until the same number of circuit evaluations is required as for Childs and Wiebe's formulas.

We have plotted the error bounds of all formulas for fixed circuit depth between all methods for one particular optimization of the multi-product formula parameters in Fig. 5. Optimizing the set of parameters $\{\boldsymbol{b}^{(r)}\}$, we tend to achieve noticeably lower resolution factors than obtained using Childs and Wiebe's multi-product formula. We generally find the matching multi-product formula to have a smaller resolution factor than the closed-form multi-product formula. While the matching multi-product formula has a better resolution, it requires an additional layer of classical optimization.

It is important to note that since we did not find useful bounds relating $\boldsymbol{b}$ to any of the resolution factors, further improvements of the bounds in Fig. 5 seem possible. In the absence of relations $\Xi(b)$, it is necessary to numerically optimize all $\boldsymbol{b}^{(r)}$ for a chosen loss function, which is not the case for Childs and Wiebe's multi-product formula due to existing, analytical relations. We found that global optimization using basin hopping combined with Nelder-Mead optimization yields the best results since the optimization landscape exhibits a large number of local minima. The optimization is also sensitive to the initial guess for those parameters, with an equal spread of positive and negative integers, i.e.,

$\boldsymbol{b}_{\text{init}} = (1, -1, 2, -2, 3, -3, \ldots, \chi R + 1)$, leading to better results. It is also fruitful to vary the loss function used for optimization. On the one hand, using the error bound for a fixed τ yields the lowest error, with the downside that the corresponding resolution factors are too large to be practical. On the other hand, solely optimizing for the resolution factor might result in resolution factors arbitrarily close to one with the downside of significantly worse error bounds. We, therefore, found the most fruitful loss function to be the error bound with the modification of using the resolution factor Ξ directly instead of γ and amplifying its impact on the loss function by using Ξ^p for some power p, which allows us to balance both quantities. Nevertheless, the use of different loss functions could be explored further. We have also found that bounding each parameter b by the total maximum leads to more robust results.

To corroborate the functioning of the newly proposed multi-product formulas beyond theoretical bounds, we also compare the actual performance with that of the conventional multi-product formula and Trotter-Suzuki. Here, we compare the actual operator distance to the ideal time evolution for the following five physically plausible and meaningful Hamiltonians, with the results shown in Fig. 6: First, we consider a standard *Heisenberg Hamiltonian* with periodic boundary conditions, described by

$$H_{\text{Heisenberg}} = -\sum_{\langle i,j \rangle} (X_i X_j + Y_i Y_j + Z_i Z_j) + 2 \sum_i X_i. \tag{87}$$

This is a Hamiltonian that plays an important role in condensed matter physics, as a prototypical model capturing ferromagnetism. Trotter-Suzuki and Childs-Wiebe formulas perform extremely well, as the Hamiltonian comprises many commuting terms. Since their performance guarantees rely on nested commutators [11], this behavior is to be expected and demonstrates the superior performance of these well studied formulas for lattice Hamiltonians, which is not reached by the newly proposed multi-product formulas. However, they are less optimal for Hamiltonians with fewer commuting terms. Motivated by these findings, we now turn to investigate a Hamiltonian comprising mutually anti-commuting terms, defined as

$$H_{\text{anti}} = \sum_{i=0}^{7} Z^{\otimes i} \otimes (X + Y) + Z^{\otimes 8}. \tag{88}$$

Anti-commuting terms play an important role when using quantum simulation algorithms to investigate *fermionic quantum models*. Here, we find a notable and significant advantage of the newly proposed multi-product formulas over Trotter-Suzuki and Childs-Wiebe already for quite large parameters of $\tau = \sum |h_k| t$. Consequently, we expect our formulas to work well and exceed previous methods for fermionic systems, which, once transformed into qubit Hamiltonians using the Jordan-Wigner transformation, will have fewer commuting terms than standard lattice spin Hamiltonians.

This behavior is confirmed and further corroborated by our results of the *Sachdev–Ye–Kitaev (SYK)* model as defined in Ref. [38], whose Hamiltonian is given by

$$H_{\text{SYK}} = \frac{1}{4 \cdot 4!} \sum_{p,q,r,s=0}^{N-1} J_{p,q,r,s} \gamma_p \gamma_q \gamma_r \gamma_s, \tag{89}$$

where N is the number of Majorana fermion mode operators γ_p and the $J_{p,q,r,s}$ are real-valued scalars drawn randomly from a normal distribution with variance $\sigma^2 = 3!/N^3$. This is an intricate local, but not geometrically local, model that is believed to provide insights into instances of strongly correlated quantum materials. It is used in the study of scrambling dynamics and has a close relation with discrete models that capture aspects of holography in the black hole context.

Again invoking the Jordan-Wigner transformation, N Majorana fermion mode operators can be mapped onto $N/2$ qubits. For our simulations, we thus chose $N = 10$ and $N = 14$, leading to a five and seven-qubit model, respectively. Here, we again find a notable and in instances substantial advantage for large τ and even for large times t. This is a physically highly plausible and interesting model for which our new simulation methods fare well.

As a further fermionic system, we look at the spinful Hubbard model on two by two sites, defined in Ref. [39] as

$$H_{\text{Hub}} = -t \sum_{\langle i,j \rangle, \sigma} (a^\dagger_{i,\sigma} a_{j,\sigma} + a^\dagger_{j,\sigma} a_{i,\sigma}) + U \sum_i a^\dagger_{i,\uparrow} a_{i,\uparrow} a^\dagger_{i,\downarrow} a_{i,\downarrow} - \mu \sum_i \sum_\sigma a^\dagger_{i,\sigma} a_{i,\sigma} - h \sum_i (a^\dagger_{i,\uparrow} a_{i,\uparrow} - a^\dagger_{i,\downarrow} a_{i,\downarrow}), \tag{90}$$

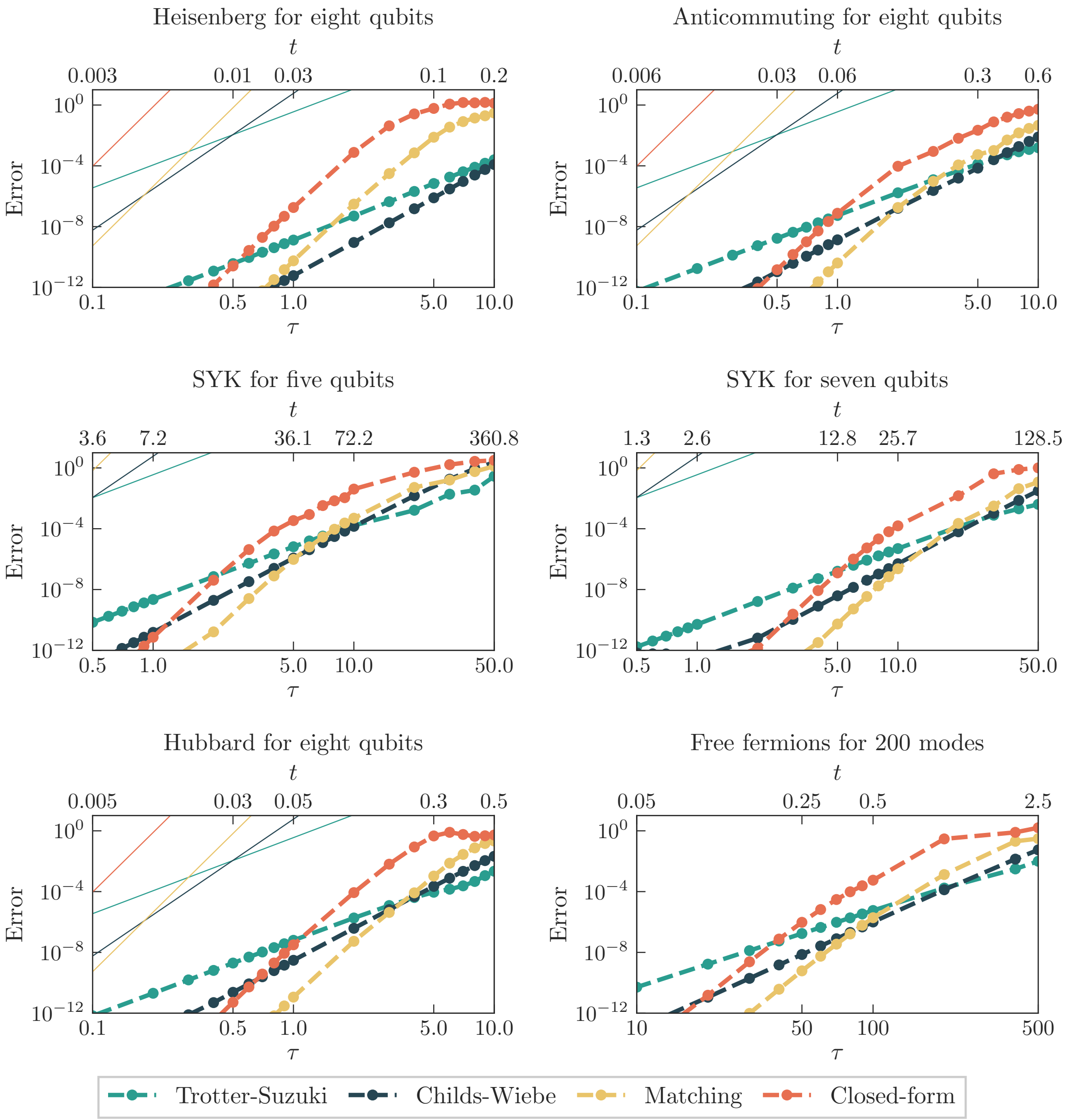


Figure 6: Numerical comparison of Trotter-Suzuki product formulas, Childs and Wiebe's multi-product formula, and the multi-product formulas proposed in this work for different $\tau = \sum_k ||h_k||t$. Here, we approximate the time evolution operator and plot the operator distance between the approximation and the ideal evolution operator for several physically plausible and interesting local, but not necessarily geometrically local, Hamiltonians, defined in equations (87), (88)m (89), (90) and (91). The simulated distances (thick, dashed lines) are much smaller than the theoretical bounds (thin, solid lines), but do have remarkably similar features overall.

with spin σ, tunneling amplitude $t = 2$, Coulomb potential $U = 2$, magnetic field $h = 0.5$ and chemical potential $\mu = 0.25$. Also, a and $a^\dagger$ represent fermionic annihilation and creation operators. These are comparably small system sizes, but already show the substantial potential of the proposed simulation method.

Finally, as a last family of examples, in order to gauge the performance of our methods for larger system sizes, we investigate a system of 200 non-interacting ("free") fermions with nearest neighbor interactions and periodic boundary conditions

$$H_{\text{ff}}(h) = \sum_{i,j} h_{i,j} a_i^\dagger a_j^\dagger, \tag{91}$$

with $h_{i,j} = 1$ if the respective fermions are nearest neighbors and $h_{i,j} = 0$ otherwise. Note that for gauging the performance for free fermions, we do not compare the time evolution operator $U(t) = \mathrm{e}^{-\mathrm{i}Ht}$ to its approximation, but the Greens function propagator $G(t) = \mathrm{e}^{-\mathrm{i}ht}$ to its approximation.

All comparisons are done in the randomized sampling framework for a Trotter-Suzuki order of $2\chi = 4$, the number of repetitions $R = 3$ and corresponding parameters for all other algorithms, ensuring an equal depth measured in the number of the required oracle calls. Although the straightforward comparison of their bounds in Fig. 5 suggests an advantage of our multi-product formulas at about $\tau = 0.1$, we find that this is the case for much larger τ already on actual systems as shown in Fig. 6. Note also that since the Hamiltonians we consider are not necessarily geometrically local, known classical simulation techniques will be heavily challenged even for comparably short simulation times

While the performance of the simple multi-product formula and Trotter-Suzuki is also much better than their bounds indicate, we find that the presented bounds for our proposed multi-product formulas are comparably looser. It is also worth noting that the performance of the matching version is slightly better than that of the closed-form version, although it comes with the penalty of an additional, classical optimization loop. We also find that the advantages of our multi-product formulas are visible even for $\tau > 1$; in the case of the SYK model, this holds even for actual simulation times $t > 1$. Additionally, we find that this advantage can also be maintained for larger system sizes, as indicated by their performance on the model of free fermions. The presented numerical studies, therefore, provide strong arguments for the functioning of our proposed multi-product formulas and their advantage in the presented regimes. It is also important to note that the corresponding resolution factors required for the randomized sampling scheme of $\Xi^{(\mathrm{cf})} \approx 1.36$ and $\Xi^{(\mathrm{m})} \approx 1.22$ are significantly better than the $\Xi^{(\mathrm{CW})} \approx 3.13$ of the multi-product formula proposed by Childs and Wiebe in Ref. [13] and would thus allow for at least four repetitions before they require the same overhead in circuit evaluations.

6 Discussion and conclusion

In this work, we have brought together two main ingredients of methods of quantum simulation. The results are notably more resource-efficient ways of performing short-time Hamiltonian simulation. These are on the one hand higher-order multi-product formulas [13, 14], on the other an element which has long been underappreciated but recently been of high interest: the element of *randomness* [20–24, 24]. Overcoming the prejudice that the time evolution has to be completed in each run of a compilation, we have introduced a novel framework for implementing multi-product formulas, to estimate expectation values of time-evolved observables.

Concretely, we have proposed a randomized sampling approach that focuses on the time evolution of the observable, not the state. When implementing multi-product formulas in a randomized fashion rather than via block encodings in the LCU framework, we can circumvent the need for additional amplitude amplification or post-selection. The results presented here have been obtained by only requiring access to a quantum-oracle machine that implements single-qubit state preparation, controlled time evolution, and quantum measurements. They are thus especially relevant in the regime of early quantum computers in which NISQ algorithms reach their limits but where full-fledged, digital, long-time evolution algorithms on fault-tolerant quantum computers are not yet available. Consequently, this work may be seen as targeting a regime in between the digital and analog setting, where we have some form of parametric control over a simulator system allowing us to compile the target time evolution with sequences of the simulator's time evolutions [40, 41]. This programmable regime then constitutes a departure from the analog setting with relatively little control over the simulator and is not as strict as digital simulation, where the control over the quantum system is strong enough to fashion its interactions into quantum gates.

Within this randomized sampling framework, we have proposed two new multi-product formulas. These schemes have been equipped with full rigorous performance guarantees. Furthermore, we have included a detailed estimation of the number of circuit evaluations that are required, a vital metric for randomized approaches. Comparing the error bounds of these newly introduced algorithms with Trotter-Suzuki product formulas and Childs and Wiebe's multi-product formula, we find that they outperform the latter for a fixed circuit depth in a practically relevant regime.

The use of multiple repetitions of short time evolutions by t/r instead of a single, long time evolution by t, elementary to achieving long simulation times, comes with a penalty in the number of circuit evaluations which scales exponentially in r. However, the base of this exponential, the resolution factor in our case, is only slightly larger than one and could be optimized even further. Even for our toy examples, it was in the range of $1.2 - 1.4$ for the newly proposed multi-product formulas as compared to 3.13 for Childs and Wiebe type multi-product formulas. Thus, at least a few repetitions are still in reach, rendering these results especially relevant for early quantum computers since we do not solely consider geometrically local Hamiltonians, for which even comparably short simulation times are a highly difficult task for known classical simulation techniques.

Benchmarking them on five different Hamiltonians, all of which are physically well motivated and each interesting in its own right, we have found that this advantage can be expected already at comparably large simulation times. While lattice Hamiltonians with many commuting terms are most likely best approximated using Trotter-Suzuki or Childs-Wiebe formulas, the newly proposed multi-product formulas show a clear advantage for fermionic Hamiltonians and those with a small number of commuting terms. This insight points to the direction that there might not be a universally optimal quantum simulation algorithm for digital quantum simulation. Instead, some algorithms could be better suited to capture the specifics of a given local Hamiltonian model. The downside of the proposed methods is that more measurements are required to reach the desired precision through the resolution factor. This resolution factor can be optimized using a classical black-box optimization, which is de facto a requirement for the functioning of the proposed multi-product formulas.

The present work is essentially bridging the gap between analog and (perhaps error-corrected), fully digital quantum technology: Not only do we expect there to be other randomized sampling schemes in digital quantum simulation, but once one can replace the element of randomness with block encodings, one can switch from these expectation value based algorithm to algorithms based on quantum phase estimation. Overall, the method introduced gives rise to a less resource-demanding way of performing Hamiltonian simulation, while also remaining conceptually and technologically simpler than for instance qubitization [18], bringing such ideas to an extent closer to the resources available in early quantum computers.

Looking ahead, it remains an open problem to relate the parameters $\boldsymbol{b}$ in Definition 1 to the resolution factor in a way that would eliminate the need for black-box optimization. So far, we can only connect the two quantities analytically, and that involves the complicated product with the Vandermonde matrix (27). Alternatively, one could improve the optimization rather than replace it. We have used only simple optimizers and loss functions, and expect possible improvements for more involved loss functions and optimization algorithms. Furthermore, the presented constructions are just two of the plethora of new multi-product formulas that could be constructed with Definition 1 and might exhibit better error bounds and resolution.

7 Acknowledgments

This work has been supported by the DFG (CRC 183 project B01 and A03, EI 519/21-1). This work has also received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No. 817482 (PASQuanS), specifically dedicated to programmable quantum simulators. It has also been supported by the BMWK (PlanQK and EniQmA), the BMBF (DAQC on notions of digital-analog quantum simulation and FermiQP on fermionic quantum processors), the Munich Quantum Valley (K8), and the Einstein Foundation (Einstein Research Unit on Quantum Devices) . M. K. acknowledges funding from ARC Centre of Excellence for Quantum Computation and Communication Technology (CQC2T), project number CE170100012. The authors endorse Scientific CO$_2$nduct [42] and provide a CO$_2$ emission table in the appendix.

References

[1] A. Acín, I. Bloch, H. Buhrman, T. Calarco, C. Eichler, J. Eisert, D. Esteve, N. Gisin, S. J. Glaser, F. Jelezko, S. Kuhr, M. Lewenstein, M. F. Riedel, P. O. Schmidt, R. Thew, A. Wallraff, I. Walmsley, and F. K. Wilhelm. "The quantum technologies roadmap: A European community view". New J. Phys. **20**, 080201 (2018).

[2] S. Lloyd. "Universal quantum simulators". Science **273**, 1073–1078 (1996).

[3] D. Aharonov and A. Ta-Shma. "Adiabatic Quantum State Generation and Statistical Zero Knowledge". arXiv:quant-ph/0301023. (2003).

[4] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders. "Efficient Quantum algorithms for simulating sparse Hamiltonians". Commun. Math. Phys. **270**, 359–371 (2007).

[5] N. Wiebe, D. Berry, P. Høyer, and B. C. Sanders. "Higher order decompositions of ordered operator exponentials". J. Phys. A **43**, 065203 (2010).

[6] N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders. "Simulating quantum dynamics on a quantum computer". J. Phys. A **44**, 445308 (2011).

[7] D. Poulin, A. Qarry, R. Somma, and F. Verstraete. "Quantum simulation of time-dependent Hamiltonians and the convenient illusion of Hilbert space". Phys. Rev. Lett. **106**, 170501 (2011).

[8] M. Kliesch, T. Barthel, C. Gogolin, M. Kastoryano, and J. Eisert. "Dissipative quantum Church-Turing theorem". Phys. Rev. Lett. **107**, 120501 (2011).

[9] R. Sweke, M. Sanz, I. Sinayskiy, F. Petruccione, and E. Solano. "Digital quantum simulation of many-body non-Markovian dynamics". Phys. Rev. A **94**, 022317 (2016).

[10] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su. "Toward the first quantum simulation with quantum speedup". PNAS **115**, 9456–9461 (2018).

[11] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu. "Theory of Trotter error with commutator scaling". Phys. Rev. X **11**, 011020 (2021).

[12] A. M. Childs and Y. Su. "Nearly optimal lattice simulation by product formulas". Phys. Rev. Lett. **123**, 050503 (2019).

[13] A. M. Childs and N. Wiebe. "Hamiltonian simulation using linear combinations of unitary operations". Quant. Inf. Comp. **12**, 901–924 (2012).

[14] G. H. Low, V. Kliuchnikov, and N. Wiebe. "Well-conditioned multiproduct Hamiltonian simulation". arXiv:1907.11679. (2019).

[15] D. W. Berry, A. M. Childs, and R. Kothari. "Hamiltonian simulation with nearly optimal dependence on all parameters". 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (2015).

[16] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma. "Exponential improvement in precision for simulating sparse hamiltonians". Proceedings of the forty-sixth annual ACM symposium on Theory of computing (2014).

[17] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma. "Simulating Hamiltonian dynamics with a truncated Taylor series". Phys. Rev. Lett. **114**, 090502 (2015).

[18] G. H. Low and I. L. Chuang. "Hamiltonian simulation by qubitization". Quantum **3**, 163 (2019).

[19] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan. "Hybrid quantum-classical algorithms and quantum error mitigation". J. Phys. Soc. Jap. **90**, 032001 (2021).

[20] E. T. Campbell. "Shorter gate sequences for quantum computing by mixing unitaries". Phys. Rev. A **95**, 042306 (2017).

[21] E. T. Campbell. "Random compiler for fast Hamiltonian simulation". Phys. Rev. Lett. **123**, 070503 (2019).

[22] A. M. Childs, A. Ostrander, and Y. Su. "Faster quantum simulation by randomization". Quantum **3**, 182 (2019).

[23] Y. Ouyang, D. R. White, and E. T. Campbell. "Compilation by stochastic Hamiltonian sparsification". Quantum **4**, 235 (2020).

[24] C.-F. Chen, H.-Y. Huang, R. Kueng, and J. A. Tropp. "Concentration for random product formulas". PRX Quantum **2**, 040305 (2021).

[25] J. Preskill. "Quantum computing in the NISQ era and beyond". Quantum **2**, 79 (2018).

[26] M. Suzuki. "General theory of fractal path integrals with applications to many-body theories and statistical physics". J. Math. Phys. **32**, 400–407 (1991).

[27] S. Blanes, F. Casas, and J. Ros. "Extrapolation of symplectic Integrators". Cel. Mech. Dyn. Astr. **75**, 149–161 (1999).

[28] S. A. Chin. "Multi-product splitting and Runge-Kutta-Nyström integrators". Cel. Mech. Dyn. Astr. **106**, 391–406 (2010).

[29] H. Yoshida. “Construction of higher order symplectic integrators”. Physics Letters A **150**, 262–268 (1990).

[30] W. Hoeffding. “Probability inequalities for sums of bounded random variables”. J. Am. Stat. Ass. **58**, 13–30 (1963).

[31] Q. Sheng. “Solving linear partial differential equations by exponential splitting”. IMA Journal of Numerical Analysis **9**, 199–212 (1989).

[32] T. A. Bespalova and O. Kyriienko. “Hamiltonian operator approximation for energy measurement and ground-state preparation”. PRX Quantum **2**, 030318 (2021).

[33] H.-Y. Huang, R. Kueng, and J. Preskill. “Predicting many properties of a quantum system from very few measurements”. Nature Phys. **16**, 1050–1057 (2020).

[34] L. Le Cam. “Locally asymptotically normal families of distributions. Certain approximations to families of distributions and their use in the theory of estimation and testing hypotheses”. Univ. California Publ. Statist. **3**, 37–98 (1960).

[35] F. S. V. Bazán. “Conditioning of rectangular Vandermonde matrices with nodes in the unit disk”. SIAM J. Mat. An. App. **21**, 679–693 (2000).

[36] M. E. A. El-Mikkawy. “Explicit inverse of a generalized Vandermonde matrix”. Appl. Math. Comp. **146**, 643–651 (2003).

[37] D.E. Knuth. “The art of computer programming: Fundamental algorithms”. Number v. 1-2 in Addison-Wesley Series in Computer Science and Information Processing. Addison-Wesley. (1973). subsequent edition.

[38] R. Babbush, D. W. Berry, and H. Neven. “Quantum simulation of the Sachdev-Ye-Kitaev model by asymmetric qubitization”. Phys. Rev. A **99**, 040301 (2019).

[39] J. R. McClean, N. C. Rubin, K. J. Sung, I. D. Kivlichan, X. Bonet-Monroig, Y. Cao, C. Dai, E. S. Fried, C. Gidney, B. Gimby, P. Gokhale, T. Häner, T. Hardikar, V. Havlíček, O. Higgott, C. Huang, J. Izaac, Z. Jiang, X. Liu, S. McArdle, M. Neeley, T. O’Brien, B. O’Gorman, I. Ozfidan, M. D. Radin, J. Romero, N. P. D. Sawaya, B. Senjean, K. Setia, S. Sim, D. S. Steiger, M. Steudtner, Q. Sun, W. Sun, D. Wang, F. Zhang, and R. Babbush. “OpenFermion: The electronic structure package for quantum computers”. Quant. Sc. Tech. **5**, 034014 (2020).

[40] S. Trotzky, Y.-A. Chen, A. Flesch, I. P. McCulloch, U. Schollwöck, J. Eisert, and I. Bloch. “Probing the relaxation towards equilibrium in an isolated strongly correlated one-dimensional Bose gas”. Nature Phys. **8**, 325–330 (2012).

[41] A. Parra-Rodriguez, P. Lougovski, L. Lamata, E. Solano, and M. Sanz. “Digital-analog quantum computation”. Phys. Rev. A **101**, 022305 (2020).

[42] R. Sweke, P. Boes, N. Ng, C. Sparaciari, J. Eisert, and M. Goihl. “Transparent reporting of research-related greenhouse gas emissions through the scientific CO2nduct initiative”. Communications Physics**5** (2022).

A CO_2 emission table

Numerical simulations	
Total Kernel Hours [h]	≈ 800
Thermal Design Power Per Kernel [W]	5.75
Total Energy Consumption Simulations [kWh]	4.6
Average Emission Of CO_2 In Germany [kg/kWh]	0.56
Total CO_2-Emission For Numerical Simulations [kg]	2.6
Were The Emissions Offset?	**Yes**